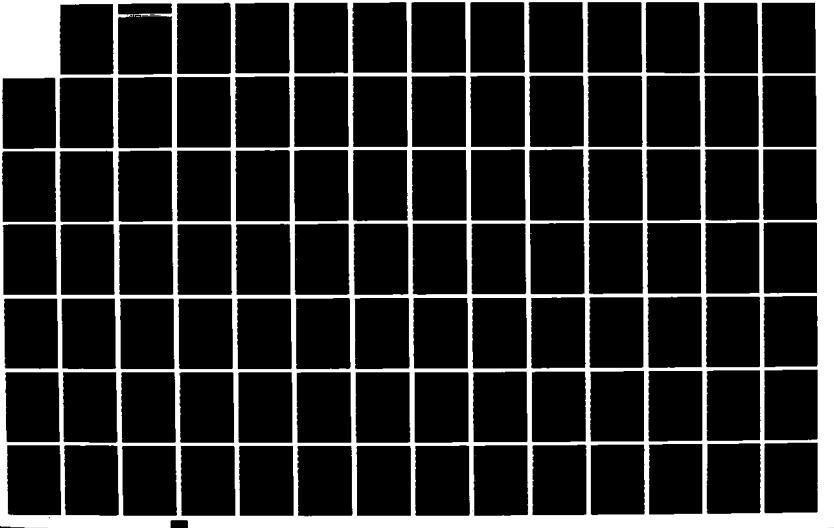
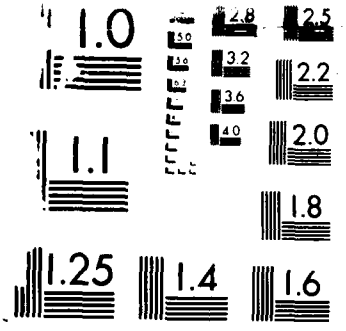
AD-A175 070 MULTISOURCE LOCALIZATION FROM DELAY AND DOPPLER MEASUREMENTS(U) SYSTEMS CONTROL TECHNOLOGY INC PALO ALTO CA B FRIEDLANDER ET AL. JUL 86 SCT-5503 1/3

UNCLASSIFIED N00014-83-C-0727 F/G 17/7 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

SCT SYSTEMS CONTROL TECHNOLOGY, INC.
1801 PAGE MILL RD. P.O. BOX 10180 PALO ALTO, CALIFORNIA 94303 (415) 494-2233

MULTISOURCE LOCALIZATION FROM DELAY AND DOPPLER MEASUREMENTS

FINAL REPORT

Julius O. Smith
Jonathan S. Abel

Systems Control Technology, Inc.
1801 Page Mill Road
Palo Alto, California, 94303

July 1986

Abstract

Project: Spectral Estimation in a Complex Environment (SPICE)
Sponsor: ONR, Contract No. N00014-83-C-0727, SCT task 5503
Personnel: B. Friedlander, J. O. Smith, and J. S. Abel
Period: 1983 to 1986

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): SCT Task 5503	5. MONITORING ORGANIZATION REPORT NUMBER(S): N00014-83-C-0727
6a. NAME OF PERFORMING ORGANIZATION: Systems Control Technology, Inc.	6b. OFFICE SYMBOL (If applicable): SCT
6c. ADDRESS (City, State and ZIP Code): 1801 Page Mill Road, Palo Alto, California 94304	7a. NAME OF MONITORING ORGANIZATION
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research	8b. OFFICE SYMBOL (If applicable): NOR
8c. ADDRESS (City, State and ZIP Code): 800 N. Quincy Street, Arlington, VA 22217-5000	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.

11. TITLE (Include Security Classification): UNCLASSIFIED
Multisource Localization From Delay and Doppler Measurements

12. PERSONAL AUTHOR(S): B. Friedlander, J. O. Smith, and J. S. Abel

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Final Report	FROM 1983 TO 1986	July 1986	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Localization, Tracking, Source location estimation, TDOA (Time-difference-of-Arrival), Time Delay, RD (Range Difference, Doppler, Spectral Estimation, Closed-form Estimators

19. ABSTRACT (Continue on reverse if necessary and identify by block number) The estimation of the locations of multiple radiating sources by analysis of passive arrray measurements is discussed. Computationally efficient, accurate source location estimators are derived based on differential delay and doppler measurements. The measurement of differential delay and doppler is also discussed; and, finally, the optimal tracking of moving sources via dynamic programming methods operating on correlogram and dopplergram data is presented.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
J. O. Smith, Y. Barniv	(415) 494-2233	

DD FORM 1473, 83 APR

Table of Contents

1. Introduction

The goal of calibrated spectral estimation is to determine sound pressure level versus frequency, at the source, versus aspect angle. Mapping measured spectra to the source and knowing aspect angle of the source require knowledge of the location and heading of the source. Furthermore, when there are multiple sources, they must be individually tracked, and their spectra must be individually resolved to the extent possible. Consequently, multi-source localization and tracking techniques comprise an important adjunct to techniques for calibrated spectral estimation.

The localization objective is carried out using acoustic signals recorded underwater by a hydrophone array. In typical passive surveillance scenarios, source location is inferred from (1) the relative arrival times and (2) the relative Doppler shifts seen among signals received at the various sensors [26].

The relative arrival time between two sensors is called the time-difference of arrival (TDOA), and a large literature on delay estimation has developed toward optimal and time-varying estimation of TDOAs (for example, a special issue on this topic appears in [21]). Delay estimation can also be applied to the echoes seen at a single sensor due to multipath; in effect, multipath reception can provide an extra "virtual sensor" located at the mirror image of a real sensor in the reflecting surface.

Estimation of differential Doppler shift is not as well studied as the delay estimation problem, and so it will probably remain a fertile research area for some time. Unlike TDOA estimation which requires a broadband signal for its accurate measurement, differential Doppler can be measured in either the narrowband or broadband case. The narrowband case is very straightforward, involving only ratios of FFT peak frequencies, while broadband spectral cross-correlation appears to offer significant opportunities for improvement. There seems to be no mature treatment of optimal broadband Doppler correlation such as exists for delay-based correlation. An advantage of localization based on Doppler is that Doppler shifts can be reliably measured at much greater range than TDOA under typical

circumstances.

When there is more than one source, localization is typically accomplished by repeated application of the single-source methods. In the case of measuring TDOAs, each source produces its own secondary peak in the cross-correlation function between two sensors; if the peaks are resolved (i.e., the signal bandwidths are sufficiently large and the TDOA times are sufficiently separated), then there is no problem obtaining N TDOAs per sensor in the case of N sources. However, it is then necessary to try, in principle, all possible associations of TDOA to source; that is, N groups must be chosen, each consisting of one TDOA from each sensor pair, and each group must have TDOAs arising from one source only. While not imposing any fundamental barriers, the association problem can vastly increase the computational burden in the multi-source case relative to the single-source case.

1.1 The SPICE Project

The purpose of the SPICE project at SCT has been to develop new techniques which aid in the detection and localization of multiple underwater sources. In this section, the major components of the project are summarized.

The first phase of the project was devoted to developing new methods for high-resolution spectrum analysis. Improvements in spectral resolution (given fixed statistical stability) translate directly to greater tracking accuracy. In many ways, this phase of the project was a follow-up effort to a prior project entitled "Multi-Target Tracking Studies" (MTS) which started in 1979 (cf. SCT Reports 5334-01,02). A mature treatment of the high-resolution techniques is presented in SCT Report 5498-04. See also SCT Report 5466-02. All SCT Reports mentioned in this report should be requested from the Adaptive Systems Department of the Advanced Technology Division of SCT.

The second phase of the SPICE project was devoted to developing, simulating, and evaluating a novel multisource tracking algorithm devised by Dr. Benjamin Friedlander. For source detection in a plane, the technique involves finding intersections (one per source) of five or more hyperplanes in 5-space. The multisource tracking algorithm is fully described in SCT Report

5466-05.

The third and final phase of the SPICE project was to develop more traditional sonar signal processing facilities based on nonparametric spectral estimation using an FPS array processor. Estimates of intersensor time delay, multipath time delay, intersensor differential Doppler, and absolute Doppler are all derived in one way or another from the FFT-based spectrum estimate. Both narrowband and broadband Doppler estimation techniques have been developed. Differential and absolute Doppler estimates have been produced from live sonar recordings. The high-resolution spectrum analysis program and the FFT-based power spectrum estimator are documented in SCT Report 5466-06B. The differential delay/Doppler techniques are described only in this document.

Side investigations carried out on the SPICE project included a new technique for delay estimation (SCT Report 5466-03) and the application of this method to the tracking of multipath delay (SCT Report 5466-04). Also, a constrained adaptive notch filter was developed to enable the real-time elimination of unwanted sinusoidal components ("Analysis and Performance Evaluation of an Adaptive Notch Filter" by B. Friedlander and J. O. Smith, SCT ATD/ASD, 1982).

The present final report for the SPICE project includes all material not appearing in prior reports, and provides a list of all reports and publications deriving from the SPICE project.

2. Guide to the Appendices

The appendices included here discuss the localization and tracking of sources using TDOA and differential doppler measurements. Improvements to currently existing systems are outlined, and several new computationally efficient localization and tracking methods are presented. The associated problem of measuring and tracking TDOA and differential Doppler values is also covered. Finally, an appendix discussing an interactive signal processing development environment is included. Below, brief descriptions of the appendices are given.

Appendix A discusses ideas for extending the Omni-Tracking System (OTS) to handle multiple sources more effectively. A wide range of alternatives is explored, from ways to use the existing system unmodified to a complete replacement of OTS by a super-powerful, model-based, track finder.

Appendix B describes a fast track solver devised by Dr. Friedlander. It uses simple features of "S-curves" measured in an absolute-Doppler path to compute track parameters. While measurement of the required features (such as minimum/maximum long-range Doppler) depends on a high signal-to-noise ratio, the track parameters can be found very quickly relative to more commonly used methods.

Appendix C documents various signal measurements which are useful for multisource localization. Most of these are currently in use, and the appendix serves as a review of some basic signal processing fundamentals. In addition, the autocorrelogram and its use in localization (described in Appendices A and B) are believed to be new ideas.

Appendix D describes newly developed Spherical Interpolation method for localizing a single source based on time-difference-of-arrival (TDOA) measurements. The method is very fast computationally and surprisingly accurate, coming close to the Cramer-Rao lower bound for unbiased estimators, as shown in Appendix E.

Appendix F discusses track parameter estimation from multipath information. Mathematical techniques analogous to those used to develop the Spherical Interpolation method are used.

Appendix G describes the kind of interactive development environment we feel is important to provide the underwater surveillance engineer. Its major feature is malleability allowing all signal processing tools to be used in interactive exploratory analysis with a complete set of signal display features and display controls. The reason for choosing an interactive programming environment is that one cannot know in advance what signal processing procedures are going to be most effective for localizing a given source. Instead, the measured data must be analyzed interactively for

features which will give the source away to an automated analysis. Another important feature of the proposed system is the ability to record interactive work for later compilation into an automated procedure. We believe such a system can increase the effectiveness per man-hour of localization software development by a large factor.

3. Conclusions

Several avenues for increasing the effectiveness of multisource localization have been explored in this project. Here we will indicate what appear to be the most promising directions for future development, based on our results.

We feel that the most powerful approach to multisource localization is the dynamic programming system which explicitly evaluates the likelihood of a source track by comparing actual sensor measurements to synthetic measurements generated by a model of the source moving along the hypothesized track. Such a framework provides for incorporation of all a priori knowledge about the source and the scenario. This method is discussed briefly in Appendix A and to a greater extent in Appendix F.

As discussed previously, relative time delay and relative Doppler shift between sensors provide the basis for present localization systems. As stated earlier, time-delay estimation is a relatively mature field, while Doppler estimation, especially in the broadband case, has not been fully analyzed in the literature. Specifically, broadband Doppler correlation methods (as discussed in Appendices C and G) need to be analyzed in a manner analogous to what Van Trees [25] has done for more conventional correlation methods. Additionally, the problem of converting instantaneous TDOA and Doppler measurements into source location and velocity estimates has not been thoroughly studied. In particular, currently, there appears to be only a few computationally efficient, accurate source location estimators from TDOA measurements (see appendices D and E), and no computationally efficient, accurate estimators of source location and velocity from instantaneous doppler or differential doppler measurements, although there are approximate solutions to the 2-dimensional special case of a source far away from a linear array [26].

The very effective Spherical Interpolation method for localizing a source from TDOA measurements (Appendices D, E, and F) warrants further study. In particular, it seems quite possible that the same basic approach can be applied to localization from differential Doppler measurements. If this is true, and if the Doppler case is as close to optimal as is the TDOA case, current practice in passive underwater surveillance will probably undergo a major revision.

4. References

4.1 SCT Reports and Corresponding Publications

[1] J. O. Smith and B. Friedlander, "Localization Based on Doppler measurements," Tech. Rep. 5466-06, Systems Control Technology, May 1984.

[2] Multi-Target Tracking Studies (MTS)," SCT Reports 5334-01, July 1980, and 5334-02, January 1980.

[3] "Optimal Instrumental Variables," SCT Report 5498-04 and SCT Report 5466-02, July 1983.

[4] B. Friedlander and J. O. Smith, "Multipath Delay Estimation," Tech. Rep. 5466-04, Systems Control Tech., Dec. 1983.

[5] J. O. Smith and B. Friedlander, "Estimation of Multipath Delay," ICASSP-84: Proc. IEEE Int. Conf. Acoust. Speech and Sig. Proc., Paper 15.9, San Diego CA, March 19-21, 1984.

[6] J. O. Smith and B. Friedlander, "Adaptive Multipath Delay Estimation," IEEE Trans. on Acoust., Speech, and Signal Proc., vol. ASSP-33, pp. 812, Aug. 1985.

[7] J. O. Smith and B. Friedlander, "High Resolution Spectrum Analysis Programs," Tech. Rep. 5466-05B, Systems Control Technology, Apr. 1984.

[8] J. O. Smith and B. Friedlander, "Adaptive Interpolated Time-Delay Estimation," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-21, no. 2, pp. 180-199, March 1985. Also presented at Asilomar, 1983. A longer version appears in SCT Report 5466-03.

[9] B. Friedlander and J. O. Smith, "Analysis and Performance Evaluation of an Adaptive Notch Filter," IEEE Trans. on Info. Theory, vol. IT-30, pp. 283-295, March 1984.

4.2 Literature References

[10] G. D. Forney, Jr., "The Viterbi Algorithm," Proc. IEEE, vol. 61, pp. 268-278, March 1973.

[11] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation From Discrete-Time Observations," IEEE Trans. on Info. Theory, vol. IT-20, pp. 591-598, Sep. 1974.

[12] D. C. Rife and R. R. Boorstyn, "Multiple Tone Parameter Estimation From Discrete-Time Observations," Bell Syst. Tech. J., vol. 55, pp. 1389-1410, Nov. 1976.

[13] J. J. Wolcin, "Maximum A Posteriori Estimation of Narrowband Signal Parameters," NUSC TM no. 791115, June 21, 1979. J. Acoust. Soc. of Amer. vol. 68, no. 1, pp. 174-178, July 1980.

[14] J. J. Wolcin, "Maximum A Posteriori Line Extraction: A Computer Program," NUSC TM no. 801042, March 20, 1980.

[15] Omni-Tracking System (OTS), ENSCO Inc., 5408A Port Royal Rd., Springfield, VA 22151.

[16] A. Papoulis, Signal Analysis,, McGraw-Hill, New York, 1977.

[17] R. E. Crochiere and L. R. Rabiner, Multirate Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[18] Digital Signal Processing Committee, ed., Programs for Digital Signal Processing, IEEE Press, New York, 1979.

[19] P. D. Welch, "The Use of Fast Fourier Transforms for the Estimation of Power Spectra: A Method Based on Time Averaging over Short Modified Periodograms," IEEE Trans. on Audio Electroacoust., vol. AU-15, pp. 70-73, June 1967.

[20] J. O. Smith and P. Gossett, "A Flexible Sampling-Rate Conversion Method," Proc. IEEE Conf. Acoust. Sp. and Sig. Proc., vol. 2, pp. 19.4.1-19.4.2, San Diego, March 1984.

[21] Special issue on time-delay estimation, IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. ASSP-29, June 1981.

[22] "Sonar Systems Engineering Memo: ADEC Subroutine Description," Heavy Military Electronics Dept., General Electric Co., Syracuse, NY 13201, June 1977.

[23] J. J. Wolcin, "Maximum A Posteriori Estimation of Narrowband Signal Parameters," NUSC TM no. 791115, June 21, 1979, J. Acoust. Soc. of Amer. vol. 68, no. 1, pp. 174-178, July 1980.

[24] J. J. Wolcin, "Maximum A Posteriori Line Extraction: A Computer Program," NUSC TM no. 801042, March 20, 1980.

[25] H. L. VanTrees, Detection, Estimation, and Modulation Theory, John Wiley & Sons, New York, 1968.

[26] E. Weinstein, "Optimal Source Localization and Tracking from Passive Array Measurements," IEEE Trans. Acost., Speech and Sig. Proc., vol. ASSP-30, No. 1, February 1982.

Appendix A - Extending the OTS System to Multiple Sources

The Omni-Tracking System (OTS) [1] carries out the localization objective for a single source. The OTS system estimates the track of a single maneuvering source based on Doppler cross-correlograms, delay cross-correlograms, or both. (The delay correlogram yields a time-difference of arrival (TDOA) versus time between each pair of sensors.)

This appendix discusses alternatives for extending the operation of OTS to multiple sources. Familiarity with the OTS is assumed.

A.1 Current Use of the OTS System in the Presence of Multiple Sources

When there are multiple sources, the operator must correctly associate correlogram peaks across time to obtain the differential Doppler or delay corresponding to the same single source in each sensor pairing. Such an association can be difficult, if not impossible, for an operator using only a display of correlogram peaks versus time.

BEARTRK is the track-producing component of OTS. When the operator passes the delay/Doppler information for a single source to BEARTRK, a nonlinear optimization is performed with respect to the track parameters. Because the optimization is nonlinear, convergence to the best estimate depends on initialization sufficiently close to the true track parameters. The use of a gradient/Newton descent method indicates that the problem is too large for exhaustive search. All such nonlinear optimization procedures carry the risk of convergence to a false local minimum.

The ADEC program [2] is used to prepare line tracks through the individual sensor spectrograms. An operator manually prunes the lines, removing unwanted lines, filling in partially missing lines, and labeling the image of each source line in each receiver. The frequency-versus-time tracks so obtained are then divided into all possible pairings (point by point) to produce differential Doppler estimates. To increase robustness, the Doppler ratios for several lines may be averaged together (since they are assumed to emanate from a single source). The Doppler ratios are then fed to BEARTRK for

localization in the usual way.

ADEC is a non-coherent tracker for sinusoidal lines in noise, and it performs line tracking in a manner somewhat similar to the coherent tracker MAPLE [3,4]. ADEC operates by tracking peaks through time in the normalized spectrogram, or LOFARGRAM. A running estimate of frequency $f_i(t)$ and frequency rate $d_i(t)$ is maintained for each line being tracked. The smoothed frequency and rate estimates are used to predict the peak frequency for the next time step, and this prediction defines the center-frequency of a search window within which the peak of the power spectrum is found. The window width is adaptive based on amplitude. The peak amplitudes are used to compute a running figure of merit for the tracked line related to the likelihood function for a sinusoid in white noise. The behavior of the incremental likelihood variable with respect to various thresholds determines when a line is detected, tracked, and declared finished.

The ADEC output is generated from the spectrogram at every sensor. For example, the lines in sensor 2 could be labeled as A2(t), B2(t), C2(t), and so on. The operator then decides which lines in each spectrogram are from the same original line, and Doppler ratios of the form A2(t)/A1(t) are formed. The Doppler ratio tracks are matched in BEARTRK by searching for the track-parameters which best predict the observed Doppler ratios.

A.2 Block-Exhaustive Search in BEARTRK

A straightforward extension of the OTS system to multiple sources would be to accept all of the correlogram peaks produced by CORAN (with little or no pruning by the operator) and to fit multiple tracks to these peaks. If BEARTRK would exhaustively search the parameter space, this extension would be straightforward. However, because the track estimation is based on gradient descent, it is typically not possible for the algorithm to explore different associations of correlogram peaks to sources. (To do so would normally require moving uphill on the error surface.) A multisource version of BEARTRK could be written which would combine coarse exhaustive search followed by iterative fine tuning by gradient descent. The result would be several locally optimal tracks among which the true tracks are assumed to lie. Only

BEARTRK need be modified using this approach to OTS extension.

A.3 Exhaustive Search over CORAN-Peak Associations (DASY/PREP).

Instead of developing an exhaustive-search version of BEARTRK, the association of CORAN peaks can be automated. In other words, the normally manual association of correlogram peaks across time (to arrive at a path corresponding to one source) can be carried out automatically. To try various associations of correlogram peaks in the multisource case, BEARTRK, as is, could be run in several passes, each of which is initialized to a different sequence of correlogram peaks versus time. Each pass of BEARTRK would produce a track estimate and a measure of fit. After all passes are complete, the tracks can be sorted by measure of fit, and the best ones chosen as track estimates.

An operator can greatly reduce the number of CORAN peak association tracks to be tried. In the automated case, some search heuristic reduction analogous to that performed by the operator is desirable. As an example, if there are 5 correlogram peaks per time step and 30 time steps, BEARTRK would be called on the order of $5^{30} \approx 10^{21}$ times if all possible associations were to be tried. If BEARTRK could produce a track estimate in one second, this process would take 30 trillion years. Thus, not only is exhaustive search of the parameter space prohibitive for individual tracks, it is also beyond present computing technology to try all possible associations of peaks to correlogram lines. Clearly, it is necessary to prune the associations and pass only a few "reasonable" possibilities to BEARTRK. One method for cutting down the search possibilities is to apply the Viterbi Algorithm [5].

A.4 The Block-Viterbi Algorithm (BVA).

The standard method for finding multiple paths through time in a computationally feasible way is called dynamic programming or the Viterbi algorithm [5]. The idea, in the first-order case, is to evaluate all paths in parallel through time, saving only the best paths at each time step for extension to the next. In the 5-peak, 30-step example above, there would be on the order of $5 \cdot 30 = 150$ path "evaluations" per source in contrast to

10^{21}. (A path is now defined as a connection of correlogram peaks through time for a given sensor pair - this is not to be confused with a track which still refers to the position and velocity history of a source through time.)

Unfortunately, the first-order Viterbi algorithm is not immediately applicable to the line-association problem. To see this, consider that the first time BEARTRK (or suitable replacement) is called to fit a track to the first correlogram peak, it has only one point to work with from each sensor pair. The track may be underdetermined. Therefore, we must be able to go to a higher order Viterbi algorithm which looks at several time steps before making a decision on path extension. A K^{th}-order Viterbi algorithm exhaustively evaluates K time steps per path extension (placing the number of path evaluations somewhere between 150 and 10^{21}). That is, the first K time steps are tested exhaustively, the winning N_s paths are extended by K points in the same way, and so on. (N_s is the number of paths "kept alive" at each step.) In the above example, if the order is K=3 and N_s=1 (one source), we get $5^3(30/3)$=1250 path evaluations for one source.

More generally, for N_p peaks per correlogram, N_t time steps, N_s sources, and order K time steps per path extension, the "block-Viterbi" algorithm (BVA) requires at most

$$\frac{N_s N_p^k N_t}{K} \qquad \text{BVA Path Evaluations}$$

Typically, this number refers to the number of incremental path evaluations. The cost of each "incremental evaluation" is the cost of updating likelihood of the whole path to include K new steps forward. By using log likelihood to evaluate the path, these updates are additive.

In practice, the number N_p^k can be replaced by N_m^k where N_m is the maximum number of correlation lags a path can change in one time step (typically only one or two).

There is still the problem that, far from CPA, the track estimate is very noisy. This suggests starting the BVA at the average CPA time and working outward into the past and future. This can be an important refinement because

the Viterbi algorithm does not reconsider its early decisions.

A.5 Lag-Limited Exhaustive Association

Since a realistic correlation path cannot change very quickly from time step to time step (say less than N_m correlation lags per time step), exhaustive search by dynamic programming can be made much more efficient when this constraint is employed.

Wolcin [4] proposes an effective strategy for incorporating a time-rate-of-change constraint in the context of using MAPLE [3,4] to track slowly changing sinusoidal frequencies in a spectrum. The idea is based on the observation that each possible initial path point expands into a small "wedge" of possible paths. For example, if $N_m = 1$ (defined above), then the number of points which must be considered at each time step grows as 1, 3, 5, 7, and so on. At each time step, only two more points are within reach of a path from a given point at time 0.

Wolcin's idea is to choose a set of starting points which are close enough together so that their "possible-path wedges" intersect half way across the total time span. For example, if there were 30 time steps and $N_m = 1$ as before, then path starting points would be chosen every 30 correlation lags, because the two wedges emanating from starting points 30 lags apart will overlap each other after 15 time steps. The supposition is that a true path will "capture" one of the expanding wedges of exhaustive search; therefore, it is not necessary to try all possible starting points to detect the presence (and latter trajectory) of a path. However, to obtain the initial trajectory of the path, the Viterbi algorithm is run backwards in time through the correlogram beginning at the final point of the captured path. As a final refinement, Wolcin recommends pursuing the path forward once again to verify that the best path from the newly found correct starting point in fact ends on the point used for the backward search. If the third pass diverges from the latter path segment picked up on the first pass, the path is rejected.

Detected paths are then removed by zeroing the correlation bins through which the path moved. This zeroing must be done in a manner matched to the

analysis used in obtaining the correlogram. If the signal or analysis bandwidth is small, for example, then the correlogram paths are characterized by wide peaks (covering multiple correlation lags). In this situation, a simple zeroing of correlogram along the path lags is not efficient because large sharp ridges will be left on either side of the zeroed path which can be interpreted as two more paths running in parallel with the first. In general, the path zeroing should consist of multiplication by an appropriate "window" function whose shape equals the inverse of the expected cross-correlation peak shape between signals received at two different sensors.

A.6 Path Association before Track Computation

All multisource tracking alternatives require a track-fitting procedure (such as BEARTRK) to be called for each path evaluation. A potential speed-up is to first determine a set of all feasible paths through the correlogram peaks. The set of feasible paths is all possible paths minus those deemed physically unrealistic. This automatic pruning step can greatly reduce the size of the search space. Note that feasibility constraints are easily incorporated into the Block-Viterbi Algorithm. The lag-limited search described in the previous subsection is one special case of feasibility constraints.

A.7 Dynamic Programming through Unprocessed Correlograms.

Instead of working with the output of CORAN, which consists of up to five correlogram peaks for each time step, a multiple-path-finding algorithm can be applied directly to the delay or Doppler correlogram for each sensor pair. This allows peaks to be tracked across time frames in which they are momentarily below the five largest peaks of the correlogram. In other words, once a path is under way, the nearest local maximum (of sufficient amplitude) in the correlogram can be taken as the continuation of the path, rather than depending on the path continuation lying among the five largest correlogram peaks. If there are time frames in practice where a correlogram peak is missing, direct peak tracking in the correlogram should be considered. Similarly, dynamic programming can be applied to the spectrogram (as in ADEC) or auto-Doppler correlogram (cf. Appendix C) for the purpose of measuring

absolute frequency or Doppler versus time.

A.8 Dynamic Programming through Noise-Equalized Hydrophone Signals

Sinusoidal signal components can be tracked by a coherent estimator, as is done by MAPLE [3,4]. Coherent processing yields a 3dB improvement in the signal-to-noise ratio (SNR), relative to non-coherent processing, for every doubling of the integration time.

The frequency estimate produced by a non-coherent frequency estimator is the (interpolated) location of the peak of the power-spectral density (PSD) estimate. The PSD estimate is normally computed as a time-average of the magnitude-squared FFT's of successive time frames [6]. The FFT length determines the height of a sinusoidal peak above the noise floor.

A coherent frequency estimator conceptually maximizes the magnitude of the inner product between the entire time signal and a sinusoid oscillating at the estimated frequency. The effective height of the sinusoidal component above the noise floor grows with the length of signal processed (3dB for every doubling of observation time). For the case of a single sinusoid in white Gaussian noise, this estimate coincides with the maximum-likelihood estimate (also the minimum-variance estimate in the non-Gaussian white noise case) [7]. The use of a coherent estimate requires enough data to reach an effective signal-to-noise ratio per bin much greater than zero. (The unaveraged magnitude-squared of the Fourier transform of Gaussian white noise has a (chi-squared) standard deviation equal to its mean [6].)

In the case of a time-varying frequency (due to Doppler), the maximum-likelihood estimate requires searching over all possible time-varying frequency histories and again maximizing the coherent inner product with the entire time signal. The MAPLE algorithm provides an approximate solution to this problem by using dynamic programming to incrementally maximize the inner product of the time signal with a sinusoid having piecewise linear frequency variation. In addition, MAPLE allows a penalty to be placed on frequency change, and it can be confined to search paths only in a small frequency interval (e.g., the known sinusoidal frequency plus and minus the maximum

expected Doppler shift). It is straightforward to include all realistic physical constraints associated with a Doppler-shifted sinusoidal underwater source. The highly restricted variation in this case makes MAPLE not nearly as computationally expensive as it is when searching a wide frequency region and/or fast frequency variation. The MAPLE algorithm in its present form can be used to greatly extend the range over which the Doppler-shift of individual sinusoidal lines can be tracked.

An important feature, not employed by MAPLE and normally employed by Doppler processing within OTS, is the total spectral correlation used in the Doppler correlogram. The Doppler correlogram is non-coherent, but uses the whole spectrum in its Doppler estimate; MAPLE is coherent, but looks only at one frequency at a time. (In cases of high SNR and multiple sources, MAPLE has the advantage of not producing "cross-terms" associated with spurious secondary correlations between the spectra of two different sources.) To recover the whole-spectrum advantage of the Doppler correlogram in a coherent tracker such as MAPLE, an extension is necessary: MAPLE can be modified to simultaneously track a set of spectral lines, all derived from a single Doppler track. Such a version of MAPLE would be exceedingly effective in cases where the true sinusoidal frequencies are known. In operation, a single value of Doppler (suitably constrained in its time variation and extent) would be optimized using dynamic programming so as to maximize the inner product of the time signal with the sum of Doppler-shifted sinusoids.

A.9 Dynamic Programming through the Track Parameter Space

The ideal application of dynamic programming is to directly evaluate the likelihood of the sensor measurements as a function of time through all possible source tracks. Searching directly through possible source tracks as opposed to correlogram peak tracks or spectrogram peak tracks allows maximum use of a priori knowledge regarding the physical constraints and characteristics. Also, the likelihood computation is being applied to the actual quantity of interest - the track - rather than some indirect manifestation of the track such as its correlogram/spectrogram image in sensor pairs. Finally, when the track space is searched directly, the track parameter estimation function of BEARTRK is fully absorbed.

The difference between this approach and that discussed in subsection A.2 is in the definition of sensor measurements which the track estimate must predict. Use of the raw sensor signals (or successive complex spectra as used by MAPLE) instead of CORAN peaks allows incorporation of all known environmental data such as signal-to-noise ratio, noise-floor shape, interference activity, and measurement stability.

Full-scale dynamic programming techniques for multisource tracking are described further in section 5 of Appendix G.

A.10 The Auto-Doppler Correlogram

The OTS system bases its track estimates on differential Doppler and/or delay between sensors. It seems worthwhile to consider other sources of information which may provide better track estimates even in the single-source case. In Appendix C, the Auto-Doppler correlogram is described. Essentially, instead of cross-correlating the spectrum in one sensor with that in another, the short-time PSD estimate in a single sensor is cross-correlated against the PSD in the same sensor at a fixed time.

We have found that for individual spectral lines (visible in the spectrogram of PSD versus time) the time of CPA can be estimated as the time of maximum intensity for the line. By choosing the CPA time of a particular line as the reference time for the auto-Doppler correlogram, the auto-Doppler correlogram becomes largely "matched" to the spectrum of the source emitting the reference line, and the peak track in the auto-Doppler correlogram is likely to provide a good estimate of the absolute Doppler shift versus time for the source whose CPA was determined. This procedure can be repeated for each distinct time of CPA for each line in each sensor. Presumably, these auto-Doppler correlograms would produce a set of absolute Doppler estimates for each source present. This information can be combined with differential Doppler estimates to sharpen the Doppler estimates. Also, the absolute Doppler information is needed in the fast track solver described in Appendix B.

A.11 Fast Estimation of Track Parameters

A further computational advantage can be obtained by using a path evaluation method which is less comprehensive than BEARTRK when the goal is to solve the path association problem across time and across sensor pairs. Once the association problem is solved, BEARTRK can be called to compute optimum track estimates from single-source path data for each source. A fast track solver is presented in Appendix B.

A.12 References

(See references at end of main body of report.)

[1] Omni-Tracking System (OTS), ENSCO Inc., 5408A Port Royal Rd., Springfield, VA 22151.

[2] "Sonar Systems Engineering Memo: ADEC Subroutine Description," Heavy Military Electronics Dept., General Electric Co., Syracuse, NY 13201, June 1977.

[3] J. J. Wolcin, "Maximum A Posteriori Estimation of Narrowband Signal Parameters," NUSC TM no. 791115, June 21, 1979, J. Acoust. Soc. of Amer. vol. 68, No. 1, pp. 174-178, July 1980.

[4] J. J. Wolcin, "Maximum A Posteriori Line Extraction: A Computer Program," NUSC TM no. 801042, March 20, 1980.

[5] G. D. Forney, Jr., "The Viterbi Algorithm," Proc. IEEE, vol. 61, pp. 268-278, March 1974.

[6] J. O. Smith and B. Friedlander, "High Resolution Spectrum Analysis Programs," Tech. Rep. 5466-05B, Systems Control Tech., April 1984.

[7] D. C. Rife and R. R. Boorstyn, "Single-Tone Parameter Estimation From Discrete-Time Observations," IEEE Trans. on Info. Theory, vol. IT-20, pp. 591-598, Sep. 1974.

Appendix B - Fast Estimation of Track Parameters

When tracking multiple sources, it may be necessary to try many different associations of spectral peaks to lines and sources. In such a case, the ability to quickly estimate accurate track parameters is essential.

B.1 A Fast Track Solver for the Single-source, Single-Velocity Case

The algorithm is a modified version of the one described in the SCT 85-371 proposal. The principal modification, suggested by R. Bliss, is the removal of the assumption that the range at CPA be comparable among the various sensors.

The algorithm works by measuring "S-curve" parameters and inferring track parameters. By "S-curve" we mean the general appearance of the Doppler vs. time observed at a single sensor when a source is passing by at a fixed speed.

In order to use the algorithm with the OTS system, it is necessary to convert from intersensor Doppler (as measured by CORAN, for example) to an estimate of absolute Doppler. Alternatively, an "auto-Doppler correlogram " (ADC) can be computed directly, wherein all spectrograms for a given sensor are Doppler-correlated against the spectrogram at CPA in that sensor; this will produce what we call an S-curve. The CPA frequencies measurable in the spectrogram can be used to convert the "cross-Doppler correlograms" (CDC) produced by CORAN into equivalent ADC's, thus providing ADC's with the higher noise immunity inherent in cross-correlations.

The output of ADEC (frequency tracks versus time) can be used directly to provide S-curves. Dividing each line by its CPA frequency and averaging those which appear to coincide (because they are from the same source) can be used to increase noise immunity. Yet another alternative is to produce absolute Doppler estimates using MAPLE.

B.2 Algorithm Description

Consider the case of N_r receivers and N_s=1 sources. The source is

assumed to be heading on a fixed course and speed without acceleration. Given the ADC's, there will exist exactly one S-curve for each sensor as shown in Fig. 1. (In the multisource case, an isolated S-curve is hypothesized by some choice of correlogram peak association over time. In the case of a single source with multiple sinusoidal lines, the S-curves can be combined by averaging.)

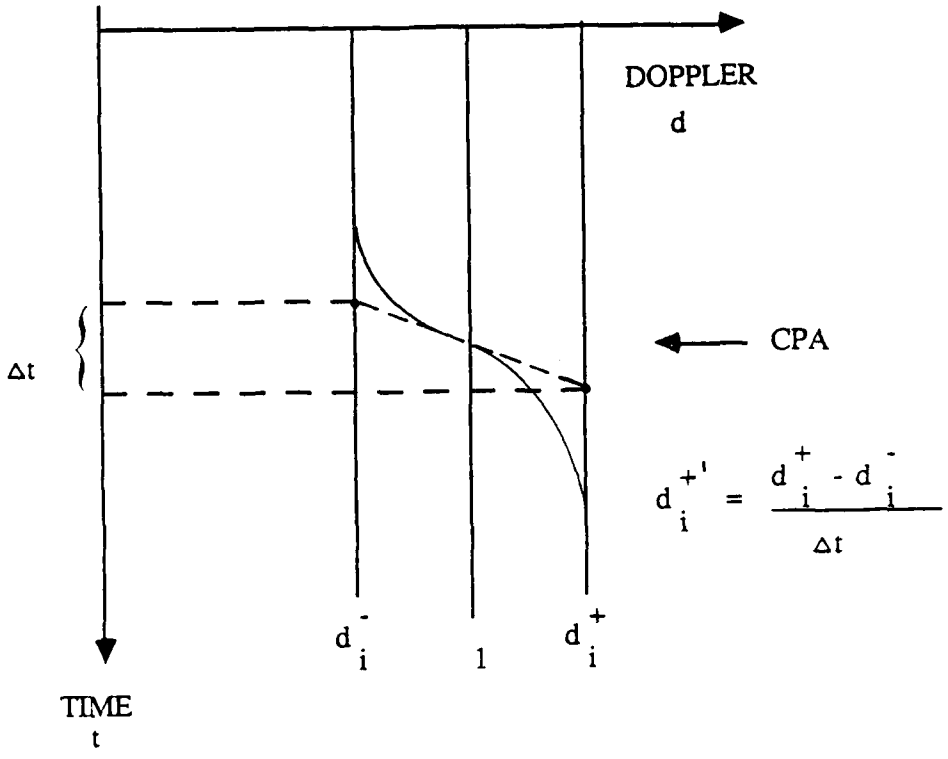


Figure 1. Doppler Shift vs. Time for a Constant-Velocity Source

The S-curve is expressed as

$$d(t) = 1 - \frac{v}{c} \cos[\theta_v(t) - \theta_i(t)] = \text{Doppler-shift vs. time} \qquad (1)$$

where v is source speed, c is sound speed, θ_i is the bearing of the source from sensor i, and θ_v is the bearing of the source track, as shown in Fig. 2.

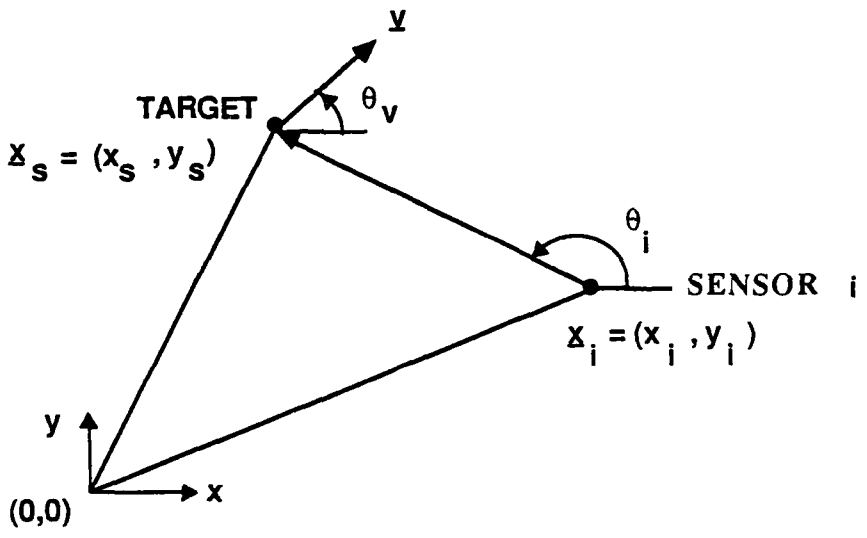


Figure 2. Source-Sensor Geometry

It is assumed that the sensor positions and the speed of sound are known constants. From the S-curve in sensor i (Fig. 1) we compute

$$d_i^+ \triangleq \max_t d_i(t)$$

$$d_i^- \triangleq \min_t d_i(t)$$

$$t_i \triangleq \text{Time of CPA at sensor i} \Rightarrow d(t_i) = 1 \tag{2}$$

$$d_i^{+\prime} \triangleq \left|\frac{\partial d_i(t)}{\partial t}(t_i)\right| = \max_s \left|\frac{\partial d_i(t)}{\partial t}(s)\right|$$

Inspection of (1) reveals that for a source speed of v the Doppler shift is between $d^+ = (1 + v/c)$ and $d^- = (1 - v/c)$. Thus, the speed of the source is estimated from each S-curve as

$$v_i = c\,\frac{d_i^+ - d_i^-}{d_i^+ + d_i^-} \tag{3}$$

and the source speed estimate is given by a simple average:

$$\hat{v} = \frac{1}{N_R} \sum_{i=1}^{N_R} v_i \tag{4}$$

Equation (1) also shows that, at CPA,

$$\dot{d}_i^{+'} = \frac{v^2}{cR_i} \tag{5}$$

where R_i is the source-receiver range at CPA for the i-th receiver. Thus, an estimate of range at CPA in each sensor is obtained as

$$R_i = \frac{1}{N_R} \sum_{i=1}^{N_R} \frac{\hat{v}^2}{c\dot{d}_i^{+'}} \tag{6}$$

From Fig. 3, we find that the angle between the track and the line between sensors i and j can be computed from

$$\sin(\gamma_{ij}) = \frac{R_j - R_i}{R_{ij}} \tag{7}$$

where R_{ij} is the distance between sensors i and j. Note that from these measurements the sign of the angle γ_{ij} is ambiguous. If ϕ_{ij} denotes the angle of the line from sensor i to sensor j, then the angle of the track can be estimated as

$$\hat{\theta}_v(i,j) \triangleq \phi_{ij} \pm \gamma_{ij} \tag{8}$$

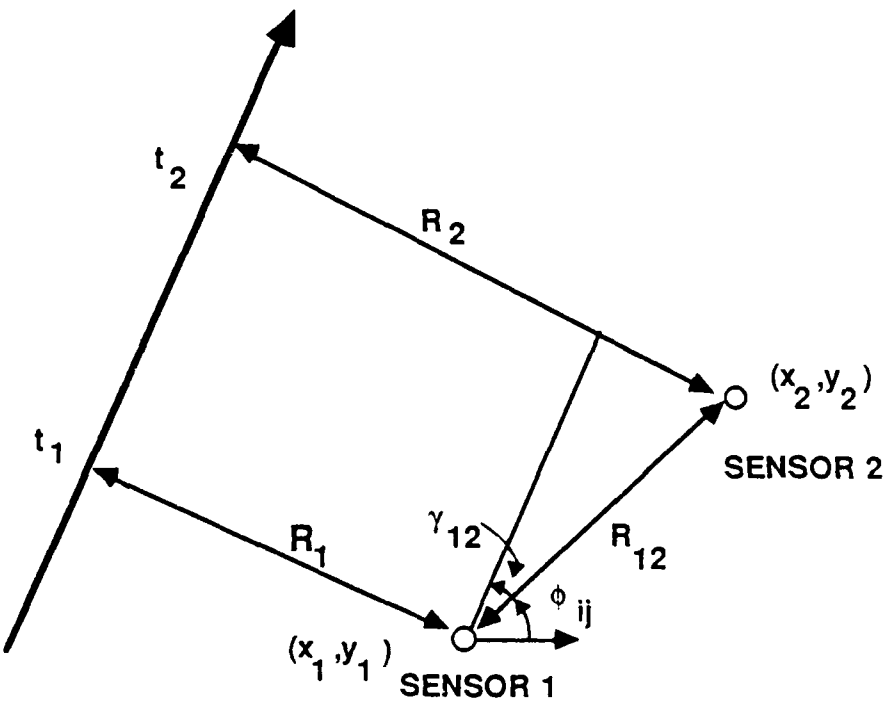


Figure 3. Two-Sensor Geometry

Averaging over all pairings, and choosing a consistent set of signs in (8), we obtain the track angle estimate as

$$\hat{\theta}_v \triangleq \frac{1}{(N_R - 1)!} \sum_{ij} \hat{\theta}_v(i,j)$$

The values of $\hat{\theta}_v(i,j)$ can be examined for outliners and pruned accordingly.

From the measured CPA times t_i, and the estimated ranges at CPA R_i, we can use the track angle estimate to find an estimate $\hat{R}_o(i)$ of the range R_o of the source from the origin (0,0) at the time of CPA τ_o with respect to the origin. The (polar) coordinate system for this is shown in Fig. 4; in these coordinates, the track is the line orthogonal to the vector from the origin to the point $(R_o, \theta_v + \pi/2)$. Let sensor i have absolute polar coordinates (r_i, ψ_i). Then

$$\hat{R}_o(i) = \hat{R}_i - r_i \sin(\theta_v - \psi_i) \tag{10}$$

The average over all sensors

$$\hat{R}_o \triangleq \frac{1}{N_R} \sum_{i=1}^{N_R} \hat{R}_o(i) \tag{11}$$

gives an estimate of the range from zero (0,0) at CPA.

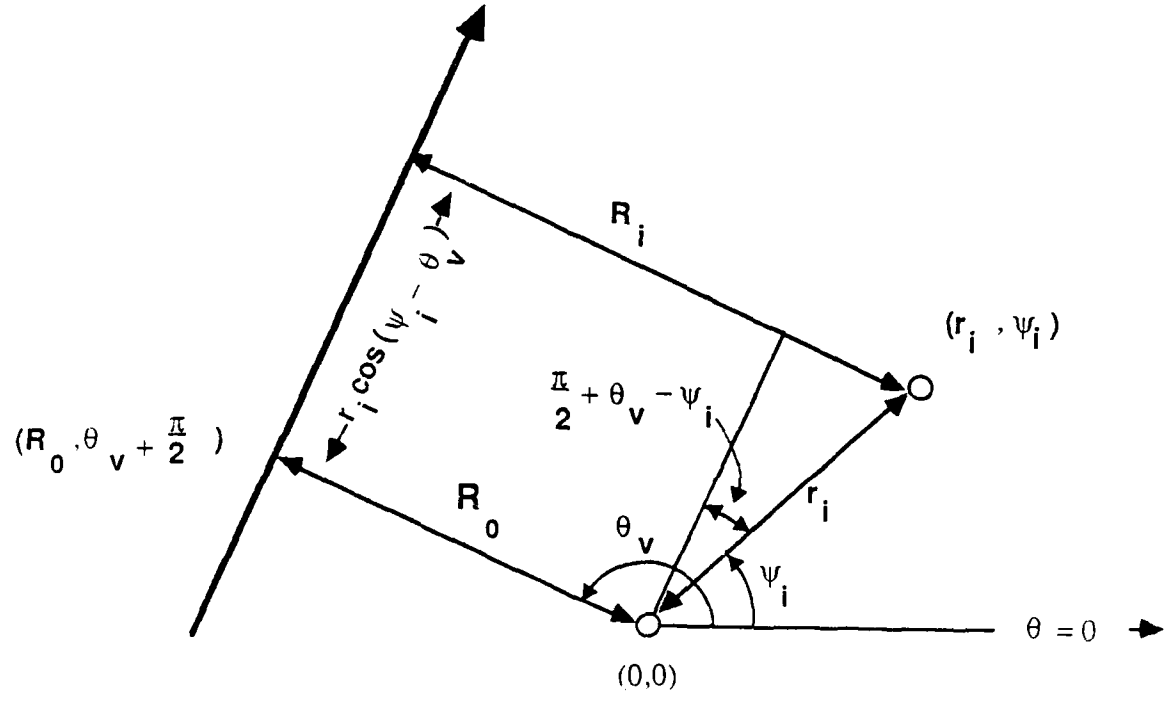


Figure 4. Absolute Polar Coordinates

Finally, given the angle $\hat{\theta}_v$ of the track and the range $\hat{R}_o$ at CPA with respect to the origin, the time of CPA to the origin τ_o can be estimated by averaging the CPA times t_i at each sensor minus the time of travel from CPA at the origin to CPA at sensor i:

$$\hat{\tau}_z = \frac{1}{N_R} \sum_{i=1}^{N_R} t_i + \frac{r_i \cos(\psi_i - \hat{\theta}_v)}{V} \tag{12}$$

The resulting parameters $(\hat{R}_o, \hat{\theta}_v, \hat{\tau}_z)$ are the track parameters output by the algorithm.

Appendix C - Measurements for Localization

In this appendix we describe measurements which are useful for multi-source localization. The original data consist of N_R digitized acoustic hydrophone signals.

C.1 Primary Measurements: Auto- and Cross-Spectra

All information is derived from the measured power spectral density (PSD) $S_{i,t}(\omega)$ in each sensor and the cross-spectral density (CSD) $S_{ij,t}(\omega)$ between each pair of sensors. The PSD is computed as described in [1], and the CSD computation is exactly analogous [2]. Briefly summarized, the PSD $S_{i,t}(\omega)$ is an average of squared-magnitude FFT's of successive blocks of digitized pressure versus time in sensor i. Similarly, the CSD is an average of the FFT at sensor i times the conjugate of the FFT at sensor j. Frequency $\omega = 2\pi f$ is in radians per second and is regarded as continuous via interpolation of FFT bins when necessary. Time t is in seconds and is defined as the mid-point of the averaging interval used to measure the PSD or CSD. Time is also available continuously through interpolation.

The time-varying PSD $S_{i,t}(\omega)$, regarded as a plot of power density versus time and frequency, with i fixed, will be called the spectrogram of the data from sensor i. Similarly, $S_{ij,t}(\omega)$ will be called the (complex) cross-spectrogram.

C.2 Secondary Measurements: Auto/Cross Delay/Doppler Correlograms

Recall that the auto(cross)-correlation function is equal to the inverse Fourier transform of the power(cross)-spectral density. Accordingly, we define the delay auto(cross) correlogram as the two-dimensional distribution obtained by replacing each slice along frequency (time fixed) of the cross-spectrogram by its inverse FFT. The term "correlogram " alone will refer to a delay cross-correlogram. The auto-correlogram can be useful for tracking multipath delays.

C.3 Differential Delay Correlograms

The correlogram between sensors i and j based on data centered at time t is given by

$$r_{ij,t}(\tilde{t}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{ij,t}(\omega) e^{j\omega\tilde{t}} d\omega$$

where $\tilde{t}$ is the correlation lag. The maximum of $r_{ij,t}(\tilde{t})$ with respect to $\tilde{t}$ is normally defined as the time-difference of arrival (TDOA) between sensors i and j.

C.4 Differential Doppler Correlograms

The second derived distribution is called the Doppler correlogram. The Doppler auto(cross)-correlogram between sensors i and j based on data centered on time t is defined by

$$D_{ij,t}(\beta) \triangleq \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{i,t}(\omega) S_{j,t}(\beta\omega) d\omega \qquad (13)$$

where β is called the differential Doppler between sensors i and j at time t. We have $D_{ij,t}(\beta) = D_{ji,t}(1/\beta)/\beta$. The Doppler auto-correlogram can be denoted more simply by $D_{i,t}(\beta) = D_{ii,t}(\beta)$.

C.5 Computation of the Doppler Correlogram

The Doppler correlogram can be efficiently computed using an FFT correlation facility [3]. The basic idea is to sample the power spectra versus log frequency in order to convert the "stretch" operation applied to $S_{j,t}(\omega)$ in (13) to a simple shift operation. This converts equation (13) into a normal cross-correlation.

The power spectra in equation (13) are transformed by the change of variable

$$\bar{\omega} = \ln\omega \qquad (14)$$

to yield

$$\bar{S}_{i,t}(\bar{\omega}) \triangleq S_{i,t}(\omega) = S_{i,t}(e^{\bar{\omega}}) \tag{15}$$

Equation (13) becomes

$$D_{ij,t}(\beta) = \int_{-\pi}^{\pi} \bar{S}_{i,t}(\bar{\omega})\bar{S}_{j,t}(\beta + \bar{\omega})e^{\bar{\omega}}d\bar{\omega} \tag{16}$$

which is simply the cross-correlation of the exponentially sampled power spectra in channels i and j, weighted by frequency $\omega = \exp(\bar{\omega})$. The weight factor can be absorbed into the power spectra so that an unweighted cross-correlation facility can be used:

$$D_{ij,t}(\beta) = \frac{1}{\sqrt{\beta}} \int_{-\pi}^{\pi} S'_{i,t}(\bar{\omega})S'_{j,t}(\beta + \bar{\omega})d\bar{\omega} \tag{17}$$

where

$$S'_{i,t}(\bar{\omega}) \triangleq e^{\bar{\omega}/2}\bar{S}_{i,t}(\bar{\omega}) = e^{\bar{\omega}/2}S_{i,t}(e^{\bar{\omega}}) = \sqrt{\omega}\, S_{i,t}(\omega) \tag{18}$$

C.6 Spectral Resampling

In practice, the power spectra are available in sampled form $S_{i,t}(\omega_k) = 2\pi k/N$, $k=0,1,2,\ldots,N_f-1$, where N_f is the number of uniformly spaced samples around the unit circle (typically the FFT size). In this situation, the exponentially sampled replacement $\bar{S}_{i,t}(\bar{\omega}_k)$ is computed using digital interpolation.

Typically, the power spectrum is assumed to be bandlimited to less than one-half the sampling frequency f_s. Unfortunately, this requires that the corresponding autocorrelation function be of infinite extent in the lag domain. Spectral interpolation is meaningful only when the corresponding correlation function asymptotically approaches zero (excluding periodic correlation functions which are easily interpolated with zeros between the original spectral samples). Therefore, proper spectral interpolation is carried out by first extending the autocorrelation function by means of

bandlimited extrapolationm [4] until the correlation function falls below the minimum representable number (i.e., it is quantized to zero beyond some lag), and then performing time-limited interpolation [5] in the frequency domain. Time-limited interpolation in the frequency domain is accomplished by zero-padding the extrapolated correlation function and taking the FFT.

A method for power spectrum interpolation which avoids bandlimited extrapolation is as follows. Assume the autocorrelation function is Bartlett windowed. That is, if $r_{i,t}(\tau)$ denotes the sample autocorrelation function for sensor i at time t, the Bartlett (or triangular) windowed autocorrelation is given by

$$\tilde{r}_{i,t}(\tau) \triangleq \begin{cases} r_{i,t}(\tau)\,\dfrac{N_f - 2|\tau|}{N_f}, & |\tau| \leq N_f/2 \\ 0, & |\tau| > N_f/2 \end{cases} \tag{19}$$

The power spectrum then is the convolution of the true power spectral density (PSD) with a sinc^2 function (defined below) having a zero-crossing interval equal to 4 spectral samples (or "bins").

When the periodogram method for PSD estimation is used [7], as we use here [1], a length $N_f/2$ rectangular window is applied to the data segment, a length N_f FFT is taken (with zero padding), and the magnitude squared is averaged over successive blocks. In this case, the true PSD is convolved with the same sinc^2 function as when Bartlett-windowing the true correlation function.

Because the periodogram method of power spectrum estimation is an average of the squared magnitude of finite-length transforms, time-limited interpolation is a valid operation (assuming sufficient zero-padding so that the operation of taking magnitude squared of the FFT results in a non-circular Bartlett-windowed sample autocorrelation).

Given periodogram-method PSD samples $S_{i,t}(\omega_k)$, $\omega_k = 2\pi l/N_f$, $k = 0,1,2,\ldots, N_f-1$, the interpolated values are defined by

$$s_{i,t}(\omega) \triangleq \sum_{k=-\infty}^{\infty} S_{i,t}(\omega_k)\,\mathrm{sinc}(\omega - \omega_k) \tag{20}$$

where

$$\mathrm{sinc}(\omega) \triangleq \frac{\sin(\omega/2)N_f}{(\omega/2)N_f} \tag{21}$$

In practice, the sinc interpolating function is normally symmetrically windowed to the first $K_s = 3$ zero crossings in each direction. If $\omega_s(\omega_k)$ denotes this window function of length $2K_s + 1$ spectral bins, then the interpolated values are approximated by

$$S_{i,t}(\omega) \approx \sum_{k=k_1}^{k_2} S_{i,t}(\omega_k)\omega_s(\omega - \omega_k)\mathrm{sinc}(\omega - \omega_k) \tag{22}$$

where

$$\omega_s(\omega) = 0.54 + 0.46\cos\left(\omega \frac{N_f}{2K_s}\right), \quad |\omega| < \frac{2\pi K_s}{N_f} \qquad \text{(Hamming window)}$$

$$k_1 \triangleq \max\{0, \omega - \omega_s\}$$

$$k_2 \triangleq \min\{N_f/2, \omega + \omega_s\} \tag{23}$$

$$\omega_s \triangleq 2\pi K_s/N_f$$

K_s = Number of zero-crossing retained per wing

If the intersample spacing is to be greater than the original sample spacing, it is desirable to smooth the power spectrum so that aliasing does not occur in the lag domain. Let $T' > 1$ denote the desired maximum bin spacing in the frequency domain. ($T' = 1$ yields the original bin spacing.) The ideal interpolation formula (20) extended to allow subsampling without aliasing is then

$$S_{i,t}(\omega) \triangleq \frac{1}{T'} \sum_{k=-\infty}^{\infty} S_{i,t}(\omega_k)\,\mathrm{sinc}\left(\frac{\omega - \omega_k}{T'}\right) \tag{24}$$

The practical windowed version is analogous to (22). An efficient table-lookup implementation of windowed-sinc interpolation is described in [6].

C.7 Exponentially Spaced Spectral Sampling

We prepare the sample PSD versus sampled log frequency $\overline{S}_{i,t}(\overline{\omega})$ by resampling the linearly spaced frequency axis to produce

$$\overline{S}_{i,t}(\overline{\omega}) \triangleq S_{i,t}(e^{\overline{\omega}})$$

It is necessary to choose the total frequency interval and sampling density over which to perform this transformation. For convenience in later processing, we specify (1) the highest frequency ω_f to be retained after mapping and (2) the number $\overline{N}_f$ of resulting samples in $\overline{S}_{i,t}(\overline{\omega})$. The lower cutoff frequency is computed from these inputs along with the maximum allowed bin spacing which is always set to the original bin spacing (to avoid loss of spectral information in the retained band).

The resulting algorithm is as follows. Let P(k) denote the array containing $S_{i,t}(\omega_k)$, $k=0,1,\ldots,K_f < N_f/2$, and let

$$\overline{P}(\overline{k}) \triangleq P(\overline{k}_0 r^{\overline{k}}), \qquad \overline{k}=0,1,\ldots,\overline{N}-1 \tag{25}$$

denote the corresponding resampled array, where k_0 is the bin coordinate of the lowest frequency retained, and $r>1$ is the ratio of the sampling interval increase. The requirement $\overline{\omega_{\overline{N}}} - \overline{\omega}_{N-1} < 2\pi/N_f$ reduces approximately to the requirement $\omega_f(1-1/r) < 2\pi/N_f$ which in conjunction with specification of ω_f and $\overline{N}$ yields

$$r \triangleq \frac{k_f}{k_f-1}$$

$$k_0 \triangleq k_f r^{-\overline{N}}$$

where $k_f < N_f/2$ is the bin coordinate of the highest frequency retained.

C.8 References

[1] J. O. Smith and B. Friedlander, "High Resolution Spectrum Analysis Programs," Tech. Rep. 5466-05B, Systems Control Tech., April 1984.

[2] Digital Signal Processing Committee, ed., "Programs for Digital Signal Processing," IEEE Press, New York, 1979.

[3] Omni-Tracking System (OTS), ENSCO Inc., 5408A Port Royal Rd., Springfield, VA 22151.

[4] A. Papoulis, Signal Analysis, McGraw-Hill, New York, 1977.

[5] R. E. Crochiere and L. R. Rabiner, Multirate Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, NY, 1983.

[6] J. O. Smith and P. Gossett, "A Flexible Sampling-Rate Conversion Method," Proc. IEEE Conf. Acoust. Sp. and Sig. Proc., vol. 2, pp. 19.4.1-19.4.2, San Diego, March 1984.

[7] Digital Signal Processing Committee, ed., Programs fr Digital Signal Processing, IEEE Press, New York, 1979.

Appendix D - The Spherical Interpolation Method of Source Localization

THE SPHERICAL INTERPOLATION METHOD OF SOURCE LOCALIZATION

J. O. Smith and J. S. Abel

Systems Control Technology
1801 Page Mill Road
Palo Alto, CA 94304

ABSTRACT

A closed-form least-squares approximate maximum likelihood method for localization of broad-band emitters from time-difference-of-arrival (TDOA) measurements, called the Spherical Interpoaltion (SI) method, is presented. The localization formula is derived from least-squares "equation-error" minimization. Computer simulation results show that the SI method has variance approaching the Cramer-Rao lower bound.

1. Introduction

The problem of automatically locating a radiating or reflecting source by analyzing signals received from the object is a basic one having numerous applications in defense, aerospace, geophysics, and industry. This paper discusses a highly efficient solution to the problem of locating a source with a passive, stationary sensor array.

A source reveals information about its location through the relative time delays seen among the signals it radiates to a sensor array. In a constant-velocity medium, the time difference of arrival (TDOA) between signals received in a sensor pair will place the source on a hyperboloid of revolution with an axis along the line drawn between the sensors. In an n-dimensional space, n such TDOA's from n nondegenerately placed sensor pairs are necessary and sufficient to uniquely determine the source location [5]. Source localization can therefore be performed using n+1 sensors placed so

that they do not lie in a subspace of dimension less than n. However, to miminize the effects of noise in the TDOA measurements, it is desirable to use $N > n+1$ sensors and fit to all of the TDOA's by minimizing some measure of inconsistency between the measured TDOA's and the TDOA's implied by the estimated source location.

While an extensive literature exists on the problem of estimating TDOA's from received signals [4], very few papers seem to be available on the problem of converting TDOA values into source location [1,2,3,5,11,12]. Conventionally, source locations are estimated by intersecting hyperbolic lines of position (LOP) determined by range difference measurements [5,11,12]. However, finding the intersection of a set of hyperboloids is computationally intensive, involving finding the minimum of a non-convex function. R. O. Schmidt [1] has proposed a formulation in which the source location is found as a focus of a conic passing through three sensors. Schmidt's method can be extended to an optimal closed-form localization technique [8]. In [2], a gradient search localization procedure is derived for computing optimal source locations from noisy TDOA's. In [3], a formula is given for single-source TDOA localization from four sensors, and can be extended to an arbitrary number of sensors [8].

In [8], a closed-form localization technique, termed the Spherical-Interpolation (SI) method, was described and was shown to perform better than two related techniques [1,3]. In this paper, we present the SI method and develop expressions for the variance of SI source location estimates in the presence of noisy TDOA measurements. We give a geometric interpretation of the source location estimates produced by the SI method and show that the SI estimates are related to maximum likelihood estimates. In addition, simulation results are presented in which the SI method shows noise immunity approaching the Cramer-Rao lower bound.

The structure of the paper is as follows. In section 2 we derive the SI method for closed-form localization of a source in a field of N sensors. In section 3, simulation results are presented for two different source locations and two additive TDOA noise levels. Section 4 gives a geometric interpretation of the SI technique.

2. Closed-Form Localization of One Source Using N Sensors

Let N denote the number of sensors, and let d_{ij}/c denote the TDOA between sensors i and j ($i,j=1,\ldots,N$) where c is the speed of propagation. The vector of (x,y,z) spatial coordinates for the i^{th} sensor is denoted $\underline{x}_i$, and the position of the source is denoted $\underline{x}_s$. The distance between the source and sensor i is denoted by $D_i = \|\underline{x}_i - \underline{x}_s\|$, and the distance from the origin to the point $\underline{x}_i$ is denoted R_i. Similarly, $R_s = \|\underline{x}_s\|$. These quantities appear in Figure 1a.

The TDOA between sensors i and j is equal to the range difference (RD) d_{ij} divided by the speed of propagation, a constant velocity medium is assumed. It will be more convenient to work directly with RD's instead of TDOA's.

The localization problem is to determine $\underline{x}_s$ given d_{ij} for i and j between 1 and N. Note that there are

$$\binom{N}{2} = \frac{N(N-1)}{2}$$

distinct RD's d_{ij} (excluding $i=j$, and counting each $d_{ij} \equiv -d_{ji}$ pair once); however, any N-1 RD measurements which form a "minimal spanning subtree" determine all the rest (in the noiseless case). The redundancy of the complete set of RD measurements is used to increase noise immunity.

We have the following basic relations:

$$d_{ij} \triangleq D_i - D_j, \qquad i = 1,\ldots,N,\ j=1,\ldots,N \tag{1}$$

$$D_i \triangleq \|\underline{x}_i - \underline{x}_s\| \tag{2}$$

$$\begin{aligned} D_i^2 &= \|\underline{x}_i\|^2 - 2\langle \underline{x}_i, \underline{x}_s \rangle + \|\underline{x}_s\|^2 \\ &\triangleq R_i^2 - 2\underline{x}_i^T\underline{x}_s - R_i^2 \end{aligned} \tag{3}$$

2.1 The Equation-Error Formulation

We assume here that the set of RD measurements d_{ij}, i=1,...,N, for some j, are available. To solve the localization problem, we first map the spatial origin to an arbitrary sensor, say the j^{th}. This gives

$$\underline{x}_j = \underline{\Delta} \underline{0} \implies \begin{cases} R_j = 0 \\ D_j = R_s \end{cases} \tag{4}$$

From (1) and (4) we have

$$D_i = D_j + d_{ij} = R_s + d_{ij} \tag{5}$$

as illustrated in Figure 1b. Substituting $R_s + d_{ij}$ for D_i in (3) yields

$$R_i^2 - d_{ij}^2 - 2R_s d_{ij} - 2\underline{x}_i^T \underline{x}_s = 0 \tag{6}$$

The j^{th} equation is degenerate so we have N-1 equations in three unknowns $\underline{x}_s$.

As the delays are typically not measured precisely, we introduce a so-called "equation error" [7] into the right-hand-side of (6), and minimize it in a least squares sense to provide an estimate of the true solution. Without loss of generality, let j=1. Then (6) becomes

$$\varepsilon_i = R_i^2 - d_{i1}^2 - 2R_s d_{i1} - 2\underline{x}_i^T \underline{x}_s , \qquad i = 2,3,\ldots,N \tag{7}$$

where ε_i is the equation error to be minimized. The set of N-1 equations (7) can be written in matrix notation as

$$\underline{\varepsilon} = \underline{\delta} - 2R_s \underline{d} - 2S\,\underline{x}_s \tag{8}$$

where

$$\underline{\delta} = \begin{bmatrix} R_2^2 - d_{21}^2 \\ R_3^2 - d_{31}^2 \\ \vdots \\ R_N^2 - d_{N1}^2 \end{bmatrix}, \quad \underline{d} \triangleq \begin{bmatrix} d_{21} \\ d_{31} \\ \vdots \\ d_{N1} \end{bmatrix}, \quad S \triangleq \begin{bmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix}, \quad \underline{\varepsilon} = \begin{bmatrix} \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_N \end{bmatrix}$$

It is worth noting that, formally, equation (8) is linear in $\underline{x}_s$ given R_s, and it is linear in R_s given $\underline{x}_s$. Error vectors which are linear in the unknowns yield closed-form least-squares solutions.

The formal least-squares solution for $\underline{x}_s$ given R_s is

$$\underline{x}_s = \frac{1}{2} S^* (\underline{\delta} - 2R_s \underline{d}) \tag{9}$$

where

$$S^* \triangleq (S^T S)^{-1} S^T \tag{10}$$

yields the unweighted least-squares solution. If it is desired to weight the RD's according to a priori confidence in each RD, then the weighted equation error energy $\underline{\varepsilon}^T W \underline{\varepsilon}$ is minimized for

$$S^* = (S^T W S)^{-1} S^T W \tag{11}$$

where W is positive definite.

To obtain a true least squares estimate for $\underline{x}_s$, it is necessary to minimize $J = \underline{\varepsilon}^T \underline{\varepsilon}$ (or $J = \underline{\varepsilon}^T W \underline{\varepsilon}$) with respect to $\underline{x}_s$ while allowing R_s to vary, maintaining the relation $R_s = \|\underline{x}_s\|$.

Unfortunately, this minimization is over a non-convex cost function, and exhaustive search techniques must be used in general. However, in this case a closed-form solution can be found which approximately minimizes $J = \varepsilon^T W \varepsilon$.

2.2 The Spherical-Interpolation Method

The basic idea of the new closed-form solution is to substitute (9) into (8) and minimize the equation error again, this time with respect to R_s. This, surprisingly, yields a linear least squares problem for finding R_s, and the solution is computationally inexpensive. The technique is made possible by the fact that the formal least squares estimate of $\underline{x}_s$ given R_s in (8) is itself linear in R_s. When the minimizing R_s value is found in this new linear equation, the corresponding value of $\underline{x}_s$ (via (9)) is automatically a minimizer of the squared equation-error norm with respect to $\underline{x}_s$ given this R_s.

Rewriting the equation error (8) to eliminate $\underline{x}_s$ by subsituting the value from (9), we get

$$\underline{\varepsilon} = \underline{\delta} - 2R_s\underline{d} - S\,S^*\,(\underline{\delta} - 2R_s\underline{d}) = (I - S\,S^*)(\underline{\delta} - 2R_s\underline{d})$$

where I is the N-1 by N-1 identity matrix. Now, $\underline{\varepsilon}$ is <u>linear</u> in the single unknown R_s. Define the N-1 by N-1 symmetric matrix

$$P_S \triangleq S\,S^* = S\,(\,S^TW\,S\,)^{-1}S^TW$$

The rank of P_S is at most 3 regardless of its order N-1. Also, P_S is <u>idempotent</u> ($P_S^2 = P_S$) . Finally, the identity matrix minus an idempotent matrix is idempotent, i.e., $(\,I - P_S\,)^2 = I - P_S$. Idempotent matrices are projection operators; the operation $P_S\,\underline{x}$ will remove components of $\underline{x}$ not lying in the space spanned by the columns of S.

In the four-sensor case, $P_S = I$, and the error $\underline{\varepsilon}$ is the zero vector. In the more general case of $N > 4$ sensors,

$$\underline{\varepsilon} = P_S^{\perp}\,(\underline{\delta} - 2R_s\underline{d}) \triangleq (I - P_S\,)(\underline{\delta} - 2R_s\underline{d}) \qquad (12)$$

so that the equation-error energy becomes

$$J = \underline{\varepsilon}^T \underline{\varepsilon} = (\underline{\delta} - 2R_s\underline{d})^T \tilde{T} (\underline{\delta} - 2R_s\underline{d})$$

where,

$$\tilde{T} \triangleq P_S^{\perp}$$

(or, to minimize $J = \underline{\varepsilon}^T W \underline{\varepsilon}$, let $\tilde{T} \triangleq P_S^{\perp} W P_S^{\perp}$). Minimizing J with respect to R_s is a form of <u>weighted least squares</u> in which the weighting matrix $\tilde{T}$ is idempotent of rank N-4. The missing dimensions reflect the degrees of freedom removed by choosing sensor 1 as the origin and substituting in the least-squares solution (9) for the three spatial source coordinates. The solution is given by

$$\tilde{R}_s = \frac{1}{2} \frac{\underline{d}^T \tilde{T} \underline{\delta}}{\underline{d}^T \tilde{T} \underline{d}} \tag{13}$$

Substituting this solution into (9) yields source location estimate

$$\hat{\underline{x}}_s = \frac{1}{2} S^* (\underline{\delta} - 2\tilde{R}_s \underline{d}) \tag{14}$$

Clearly, the computational burden of (14) is very low compared to iterative nonlinear minimization. If iterative nonlinear minimization is desired (to obtain the lowest possible variance and bias), (14) provides an excellent initial value for a general descent method.

Note that the pair $(\hat{\underline{x}}_s, \tilde{R}_s)$ minimizes the equation-error energy $J = \underline{\varepsilon}^T W \underline{\varepsilon}$ <u>without</u> the constraint $\tilde{R}_s = \|\hat{\underline{x}}_s\|$. We expect that the pair $(\hat{\underline{x}}_s, \hat{R}_s = \|\hat{\underline{x}}_s\|)$ <u>approximately</u> minimizes J subject to the constraint $\|\hat{\underline{x}}_s\| = \hat{R}_s$. Therefore, we define the <u>range estimate</u> by

$$\hat{R}_s \triangleq \|\hat{\underline{x}}_s\|$$

instead of using (13). Similarly, the <u>bearing estimate</u> is defined as the vector of direction cosines from the origin at sensor 1 to the source:

$$\hat{\underline{\Omega}}_s \triangleq \frac{\hat{\underline{x}}_s}{\hat{R}_s} = \frac{\hat{\underline{x}}_s}{\|\hat{\underline{x}}_s\|} \tag{15}$$

Note that the SI solution (14) is based on RDs measured relative to a single reference sensor. When additional RDs are available, additional SI source location estimates can be made, using different reference sensors, and the source location estimate could be computed as the weighted average of the SI estimates.

2.3 Variance of the SI Estimate

Here, we give expressions for the variance of the unweighted SI estimates of source location, range and bearing. Assuming the RD variance is small compared to the RD mean, the variance of the SI estimator of source location is given by [10]

$$\mathrm{Var}(\hat{\underline{x}}_s) \approx \left(\frac{\partial \underline{x}_s}{\partial \underline{d}}\right) \underline{R}_d \left(\frac{\partial \underline{x}_s}{\partial \underline{d}}\right)^T \tag{16}$$

where the RD vector, $\underline{d}$ is assumed unbiased with covariance matrix $\underline{R}_d$. Accordingly, the variance of the source range and source direction cosines vector are given by

$$\begin{aligned} \mathrm{Var}(\hat{R}_s) &= \left(\frac{\partial R_s}{\partial \underline{x}_s}\right) \mathrm{var}(\hat{\underline{x}}_s) \left(\frac{\partial R_s}{\partial \underline{x}_s}\right)^T \\ \mathrm{Var}(\hat{\underline{\Omega}}_s) &= \left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right) \mathrm{Var}(\hat{\underline{x}}_s) \left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)^T \end{aligned} \tag{17}$$

The derivative $\partial \underline{x}_s / \partial \underline{d}$ can be evaluated by using (8) in the noiseless case, $\underline{\varepsilon} = \underline{0}$:

$$\frac{\partial \underline{\delta}}{\partial \underline{d}} - 2 \frac{\partial \underline{R}_s \underline{d}}{\partial \underline{d}} - 2\underline{S} \frac{\partial \underline{x}_s}{\partial \underline{d}} = 0 \tag{18}$$

Recall $\underline{\delta}_i = R_i^2 - d_{i1}^2$, and

$$\frac{\partial \underline{\delta}}{\partial \underline{d}} = -2\underline{\Lambda}_{\underline{d}} \tag{19}$$

where

$$\underline{\Lambda}_{\underline{d}} \triangleq \begin{bmatrix} \underline{d}_{21} & & 0 \\ & \ddots & \\ & & \underline{d}_{i1} \\ & & \ddots \\ 0 & & \underline{d}_{N1} \end{bmatrix}$$

The derivative $\frac{\partial R_s \underline{d}}{\partial \underline{d}}$ may be evaluated using the chain rule:

$$\frac{\partial R_s \underline{d}}{\partial \underline{d}} = \underline{d}\left(\frac{\partial R_s}{\partial \underline{d}}\right) + R_s \frac{\partial \underline{d}}{\partial \underline{d}} = \underline{d}\ \underline{\Omega}_s^T\left(\frac{\partial \underline{x}_s}{\partial \underline{d}}\right) + R_s I \tag{20}$$

where $R_s \triangleq \|\underline{x}_s\|$ and $\underline{x}_s/R_s = \underline{\Omega}_s$ have been used to evaluate $\partial R_s/\partial \underline{x}_s$. Using (19) and (20) in (18) we find,

$$\Lambda_{\underline{d}} + R_s I + \left(\underline{d}\ \underline{\Omega}_s^T + S\right)\frac{\partial \underline{x}_s}{\partial \underline{d}} = 0$$

and,

$$\frac{\partial \underline{x}_s}{\partial \underline{d}} = -\left(\Delta^T\Delta\right)^{-1}\Delta^T\left(\Lambda_{\underline{d}} + R_s I\right) \tag{21}$$

where

$$\Delta \triangleq \left(\underline{d}\ \underline{\Omega}_s^T - S\right)$$

Using the definitions for $\underline{\Omega}_s$ and R_s, we have,

$$\frac{\partial \underline{x}_s}{\partial R_s} = \underline{\Omega}_s$$

and

$$\frac{\partial \underline{x}_s}{\partial \underline{\Omega}_s} = \frac{1}{R_s} P^{\perp}_{\underline{x}_s} \triangleq \frac{1}{R_s}\left(I - \underline{\Omega}_s\underline{\Omega}_s^T\right)$$

where $P^{\perp}_{\underline{x}_s}$ represents a projection operator which removes components of a vector along the direction of $\underline{x}_s$. Substituting equations (21) and the above into (16) and (17), we have the desired result:

$$\begin{aligned}
Var(\hat{\underline{x}}_s) &= \left(\Delta^T\Delta\right)\Delta^T\left(\Lambda_{\underline{d}} + R_s I\right)R_d\left(\Lambda_{\underline{d}} - R_s I\right)\Delta\left(\Delta^T\Delta\right)^{-1} \\
Var(R_s) &= \underline{\Omega}_s^T\, Var(\hat{\underline{x}}_s)\underline{\Omega}_s \\
Var(\underline{\Omega}_s) &= \frac{1}{R_s^2} P_{\underline{x}_s}\, Var(\hat{\underline{x}}_s) P^{\perp}_{\underline{x}_s}
\end{aligned} \tag{22}$$

3. Simulation Results

This section presents simulation results on the performance of the SI method on the problem of localizing a single source in the presence of noise. The simulations were implemented in the Ctrl-C* language on a Vax 11/780 computer. The performance of the SI localizer is evaluated according to the bias and variance of the source location estimate. The results show that the variance of the SI method is given accurately by (22), and that performance of the SI method is nearly optimal, i.e., the expected square error in the source location estimate approaches the Cramer-Rao lower bound.

All simulations employed the 9-sensor array geometry shown in Figure 2, with the two source locations shown and two levels of additive white noise in the RD's. The sample bias and variance were obtained by averaging the results of 1000-trial Monte-Carlo runs.

Typically, RD estimates are zero mean with variance dependent only on the signal-to-noise ratio (SNR) at each sensor [4,6]. When the source-to-sensor range is comparable in all sensors, the SNR is normally comparable also. Our simulated range difference estimates were generated by adding white Gaussian noise to true RD values, corresponding to the case of uniform SNR across sensors.

---

* Ctrl-C is a registered trademark of SCT.

Table 1. Enviromental information for the four simulation cases. (Note: all distances are measured in meters.)

Run 1 --- Source location $\underline{x}_s$: (390,160,170)
Range R_s: 454.5
Bearing cosines $\underline{\alpha}_s$: (.858,.352,.374)
RD noise standard deviation $\sigma_{d_{i1}} = 0.1$

Run 2 --- Source location $\underline{x}_s$: (390,160,170)
Range R_s: 454.5
Bearing cosines $\underline{\alpha}_s$: (.858,.352,.374)
RD noise standard deviation $\sigma_{di1} = 1.0$

Run 3 --- Source location $\underline{x}_s$: (540,1360,110)
Range R_s: 1467
Bearing cosines $\underline{\alpha}_s$: (.368,.927,.075)
RD noise standard deviation $\sigma_{d_{i1}} = 0.1$

Run 4 --- Source location $\underline{x}_s$: (540,1360,110)
Range R_s: 1467
Bearing cosines $\underline{\alpha}_s$: (.368,.927,.075)
RD noise standard deviation $\sigma_{d_{i1}} = 1.0$

Table 1 describes the enviromental information (source location, range, bearing, and additive noise level) for each of four 1000-trial Monte-Carlo runs (two source locations and two TDOA noise levels). For all runs, the nine sensors were located at (0,0,0), (0,0,100), (0,0,200), (100,0,0), (100,0,100), (100,0,200), (0,100,0), (0,100,100), and (0,100,200) meters.

Table 2. Sample bias measurements, SI Method 1000 trials

Run	Source Location			Range	Source Bearing ($\times 10^4$)		
	$x_s-\hat{x}_s$	$y_s-\hat{y}_s$	$z_s-\hat{z}_s$	$R_s-\hat{R}_s$	$\Omega_x-\hat{\Omega}_x$	$\Omega_y-\hat{\Omega}_y$	$\Omega_z-\hat{\Omega}_z$
1	0.099	0.019	0.025	0.10	-.019	-.50	-.46
2	3.73	10	0.81	3.84	2.2	-12	-22
3	0.95	2.5	0.042	2.7	-.60	-.22	-1.4
4	53.	143.	1.8	153.	-35	2.4	-96

Table 2 shows the sample bias for each Monte-Carlo run, listing errors in the Euclidean 3D source coordinates and errors in range. We find that the SI method produces estimates with a bias which is small compared to the quantities being estimated. Note that the bias in the source location and range is positive, indicating that the source estimate is generally closer to the origin than the true source. Also, the bias appears to be strong function of additive noise level, perhaps increasing quadratically with an increase in noise standard deviation.

Table 3. Sample standard deviation measurements, SI Method 1000 trials

	Run	Source Location			Range	Source Bearing ($\times 10^4$)		
		$\sigma_{\hat{x}_s}$	$\sigma_{\hat{y}_s}$	$\sigma_{\hat{z}_s}$	$\sigma_{\hat{R}_s}$	$\sigma_{\hat{\Omega}_x}$	$\sigma_{\hat{\Omega}_y}$	$\sigma_{\hat{\Omega}_z}$
Sample	1	1.9	0.66	0.42	2.0	5.6	7.0	9.5
Standard	2	19.	6.5	4.1	20.	56	70	95
Deviation	3	9.5	26.	0.58	27.	7.8	6.4	13
	4	82.	220.	5.2	235.	78	63.	126
Standard	1	1.7	0.62	0.30	1.8	4.5	5.0	8.7
Deviation	2	17.	6.2	3.0	18.	45.	50.	87.
from (22)	3	9.3	25.	0.44	26.	6.2	3.0	12.
	4	93.	248.	4.4	264.	62.	30.	123.

Table 3 shows the sample standard deviations for each Monte-Carlo run, again for both Euclidean and plane-projected polar coordinates and range. Due to the source-sensor geometry, the bearing is more accurately estimated than the range in all examples; this is typical when the source is several aperture sizes away from the sensor array. Note the estimate variance of the SI method appears to increase linearly with an increase in RD variance.

Table 3 also shows the standard deviation predicted by (22). Note the agreement with the Monte-Carlo simulations.

Table 4. Sample RMS error measurements, 1000 trials.

Method	Run	Source Location			Range	Source Bearing ($\times 10^4$)		
		$\overline{\sigma}_{\hat{x}_s}$	$\overline{\sigma}_{\hat{y}_s}$	$\overline{\sigma}_{\hat{z}_s}$	$\overline{\sigma}_{\hat{R}_s}$	$\overline{\sigma}_{\hat{\Omega}_x}$	$\overline{\sigma}_{\hat{\Omega}_y}$	$\overline{\sigma}_{\hat{\Omega}_z}$
	1	1.9	0.66	0.42	2.0	5.6	7.0	9.6
SI	2	19.	6.6	4.2	2.2	56	71	98
	3	9.6	26.	0.58	27.	7.8	6.4	13
	4	98.	263.	5.5	280.	85.	63.	159.
	1	1.7	0.62	0.30	1.7	4.4	5.0	8.6
C-R	2	17.	6.2	3.0	17.	44.	50.	86.
Bound	3	9.3	25.	0.44	26.	6.2	3.0	12.
	4	93.	247.	4.4	263.	62.	30.	123.

The performance of the SI estimator can be evaluated by comparison to the Cramer-Rao (C-R) lower bound, a lower bound on the variance of any unbiased estimator [10]. Since the SI method is biased, the root-mean-squared (RMS) error, defined by

$$\overline{\sigma} \triangleq \sqrt{\text{Bias}^2 + \text{Variance}}$$

is compared to the C-R bound. Table 4 lists the Cramer-Rao lower bound standard deviation [9] along with the RMS error for the SI estimator. As seen in this table, the performance of the SI estimator is nearly optimal, approaching the C-R bound. Note that the SI estimator appears to be relatively unbiased, as the RMS error is nearly equal to the standard deviation in all cases. Therefore, we expect (22) to accurately predict the RMS error.

The extent to which the SI method is not an optimal least squares method is a function of the extent to which $\tilde{R}_s$ computed by (13) is not equal to $\|\hat{\underline{x}}_s\|$ computed by (14). It was therefore of interest to compare these quantities. In the high-noise case ($\sigma_{d_{i1}} = 1$) above, for both source positions, the quantity $|1 - \|\hat{\underline{x}}_s\|/\tilde{R}_s|$ was typically less than 0.005 and

almost never greater than 0.01. Thus, the SI method is very close to an optimal least-squares equation-error method under the above conditions. It would be of interest to find out when and if the disparity between $\bar{R}_s$ and $\|\hat{\underline{x}}_s\|$ becomes large, and, if so, what effect this has on the source location estimate.

4. Geometric Interpretation

This section gives a geometric interpretation of the SI solution and discusses it in light of the Monte-Carlo results presented above.

4.1 The Error Criterion

The goal is to localize the source $\underline{x}_s$ and therefore to minimize $\|\underline{x}_s - \hat{\underline{x}}_s\|$ for some norm. We will only consider L_2 (sum-of-squares) norms. Since only the RD measurement vector $\underline{d}$ is known, a natural formulation would appear to be

$$\underset{\hat{\underline{x}}_s}{\text{Minimize}} \ \|\underline{\varepsilon}\| \triangleq \|\underline{d} - \hat{\underline{d}}(\hat{\underline{x}}_s)\| \tag{23}$$

where $\hat{\underline{d}}(\hat{\underline{x}}_s)$ is the vector of all measurable RDs corresponding to the source location estimate $\hat{\underline{x}}_s$. This error criterion is especially well-suited to the case of zero-mean errors in the RD's. Indeed, if the RD errors are Gaussian perturbations, then (23) provides the maximum likelihood estimate for $\hat{\underline{x}}_s$ (which is now regarded as a parameter determining the mean $\hat{\underline{d}}(\underline{x}_s)$ of the multivariate normal distribution for $\underline{d}$). For this reason, the solution to (23) will be referred to as the maximum likelihood estimate of the source location.

As mentioned in the introduction, solving (23) requires nonlinear minimization techniques. For this reason, the SI method does not solve (23). Instead, it approximately minimizes the L_2 norm of an "equation error" which was chosen purely to simplify the solution. As seen in the Monte-Carlo results, good estimates are obtained nonetheless.

Placing sensor 1 at the origin as before, and using only N-1 RD's, all

referred to sensor 1, (23) can be interpreted as finding the sphere, passing through sensor 1, whose surface is as close as possible to being d_{i1} away from the i^{th} sensor. This arrangement is shown in Figure 3 for the noiseless case.

As seen in Figure 4, the sphere of radius $\hat{R}_s$, centered at $\underline{\hat{x}}_s$ and passing through sensor 1, is a surface of zero RD to sensor 1. The distance from the sphere to sensor i is then $\hat{d}_{i1}$. The problem is then to position $\underline{\hat{x}}_s$ such that $\hat{d}_{i1} \approx d_{i1}$ for every i. Accordingly, the error minimized in (23) is equivalent to the sum of squared differences between the sphere-to-sensor distance $\hat{d}_{i1}$ and the measured RD d_{i1} for that sensor.

Since the sphere around the source must always pass exactly through sensor 1, the solution is sensitive to the choice of sensor 1. Improved results in the case of a systematic bias in measuring the RD's relative to sensor 1 may be obtained by removing the constraint that the source sphere must pass through sensor 1. This can be accomplished by adding a constant γ to the RD vector, and solving for γ as well as $\hat{R}_s$ and $\underline{\hat{x}}_s$.

4.2 Spherical Interpolation

We first show that the equation error (7) (approximately minimized by the SI method) is closely related to the maximum likelihood error minimized in (23). Adding and subtracting R_s^2 in the definition of the equation error (7) gives (upon introducing hats to denote estimated quantities)

$$\varepsilon_i = R_i^2 - 2\underline{x}_i^T\underline{\hat{x}}_s + \hat{R}_s^2 - (d_{i1}^2 + 2\hat{R}_s d_{i1} + \hat{R}_s^2), \qquad i = 2,3,\ldots,N$$

$$\varepsilon_i = \|\underline{x}_i - \underline{\hat{x}}_s\|^2 - (\hat{R}_s + d_{i1})^2 \tag{24}$$

where $\underline{\hat{x}}_s$ and $\hat{R}_s$ are the estimated source location and range, $\underline{x}_i$ is the i^{th} sensor location (known exactly), and d_{i1} is the measured range difference. Let

$$\hat{d}_{i1} = \|\underline{x}_i - \underline{\hat{x}}_s\| - \hat{R}_s$$

denote the RD predicted by the source location estimate $\hat{\underline{x}}_s$ (cf. Figure 4). Then (24) can be written as

$$\begin{aligned}\varepsilon_i &= (\hat{R}_s + \hat{d}_{i1})^2 - (\hat{R}_s + d_{i1})^2 \\ &= 2\hat{R}_s(d_{i1} - \hat{d}_{i1}) + d_{i1}^2 - \hat{d}_{i1}^2 \\ &= (d_{i1} + \hat{d}_{i1} + 2\hat{R}_s)(d_{i1} - \hat{d}_{i1})\end{aligned} \tag{25}$$

Assuming that the noise in the delay estimates is small compared to the delay values, ε_i can be written as

$$\varepsilon_i \approx 2(\hat{R}_s + d_{i1})(d_{i1} - \hat{d}_{i1})$$

This form of the error displays the equation error as the maximum likelihood error $d_{i1} - \hat{d}_{i1}$ times the term $\hat{R}_s + d_{i1}$. When the source range is large compared to the intersensor separation, the SI equation error reduces to

$$\varepsilon_i \approx 2\hat{R}_s(d_{i1} - \hat{d}_{i1})$$

The difference between the SI method and the maximum likelihood method is the tendency of this error to pull the estimated source location toward the origin, thereby making $\hat{R}_s$ and the equation error smaller. This contraction of the source estimate toward sensor 1 was consistently observed in the simulations. Note that there is no difference between the maximum likelihood solution and the SI method with respect to bearing estimation when $R_s \gg |d_{i1}| + |\hat{d}_{i1}|$.

The above discussion leads naturally to a weighting function and an iterative technique for obtaining the solution to (23). The weighting matrix, W , in (11) should be given by,

$$W = \text{diag}\left[\frac{1}{d_{i1}+\hat{d}_{i1}+2\hat{R}_s}\right] \tag{26}$$

where $\text{diag}(v_i)$ is a diagonal matrix with v_i as the element in the i^{th} row and column, and $d_{i1} + \hat{d}_{i1} + 2\hat{R}_s$ is given by a prior solution of (7), or estimated from a priori information. The solution of (23) is then found by

iteratively solving (7) with successively updated values of W using (26). Assuming this iteration converges, the weighting cancels the first term of (25) and the maximum likelihood error remains. Conditions for the convergence of this scheme to the solution of (23) need to be determined.

5. Summary

In this paper, a closed-form solution for localizing a single source in n-dimensions from TDOA information gathered from an N > n+1-sensor array was described and evaluated by Monte-Carlo simulations. It was found that the SI method exhibited an RMS localization error close to the Cramer-Rao lower bound. Finally, in support of our simulation results, we showed that the SI method is closely related to the maximum likelihood estimate for the case of Gaussian TDOA measurement errors.

6. References

[1] R. O. Schmidt, "A New Approach to Geometry of Range Difference Location," IEEE Trans. Aero. and Elec. Systems, vol. AES-8, no. 6, pp. 821--835, Nov. 1972.

[2] J. M. Delosme, M. Morf, and B. Friedlander, "A Linear Equation Approach to Locating Sources from Time-Difference-of-Arrival Measurements," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, 1980.

[3] Schau and A. Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," submitted for publication.

[4] Special issue on time-delay estimation, IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. ASSP-29, June 1981.

[5] J. P. Van Etten, "Navigation Systems: Fundamentals of Low and Very Low Frequency Hyperbolic Techniques," Electrical Commun., vol. 45, no. 3, pp. 192--212, 1970.

[6] B. Friedlander, "On the Cramer-Rao Bound for Time Delay and Doppler Estimation," IEEE Trans. Info. Theory, vol. IT-30, no. 3, pp. 575--580, May 1984.

[7] L. Ljung and T. Soderstrom, Theory and Practice of Recursive Identification, MIT Press, Cambridge MA, 1984.

[8] J. O. Smith and J. S. Abel, "Closed-Form Least-Squares Localization of Multiple Broad-Band Emitters from Time-Difference-Of-Arrival Measurements," Systems Control Technology Technical Memo 5517-01.

[9] J. S. Abel and J. O. Smith, "The Cramer-Rao Lower Bound on the Variance of Source Location Estimates from Range Difference Measurements," Systems Control Technical Memo 5517-02.

[10] D. R. Cox and D. V. Hinkley, Theoretical Statistics, Chapman and Hall, London, 1982.

[11] H. B. Lee, "A Novel Procedure for Assessing the Accuracy of Hyperbolic Multilateration Systems," IEEE Trans. Acrosp. Electron. Syst., vol. AES 11, no. 1, January 1975, pp 2-15.

[12] N. Marchand, "Error Distributions of Best Estimate of Position from Multiple Time Difference Hyperbolic Networks," IEEE Trans. Aerosp. Navig. Electron., vol. ANE-11, no. 2, June 1964, pp. 96-100.

Figure Captions

Figure 1. Diagrams illustrating notation and certain geometric relations for the case of a single source $\underline{x}_s$. Labels imbedded within a line denote the length of the corresponding vector. For example, $D_i = \|\underline{x}_i - \underline{x}_s\|$.

Figure 2. Source-sensor geometry used in all simulations.

Figure 3. Sensor array, source and RD geometry. The sphere of radius R_s drawn around the source is the surface of zero RD relative to sensor 1. The perpendicular distance from the sphere to any sensor is the RD for that sensor relative to sensor 1.

Figure 4. Geometric representation of the relationship given in equation (7).

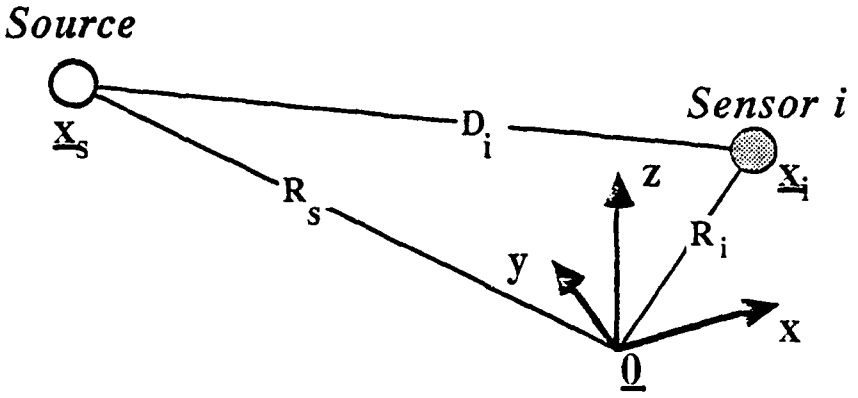

Source
Sensor i
$\underline{x}_s$
D_i
$\underline{x}_i$
R_s
z
R_i
y
x
$\underline{0}$

Figure 1a

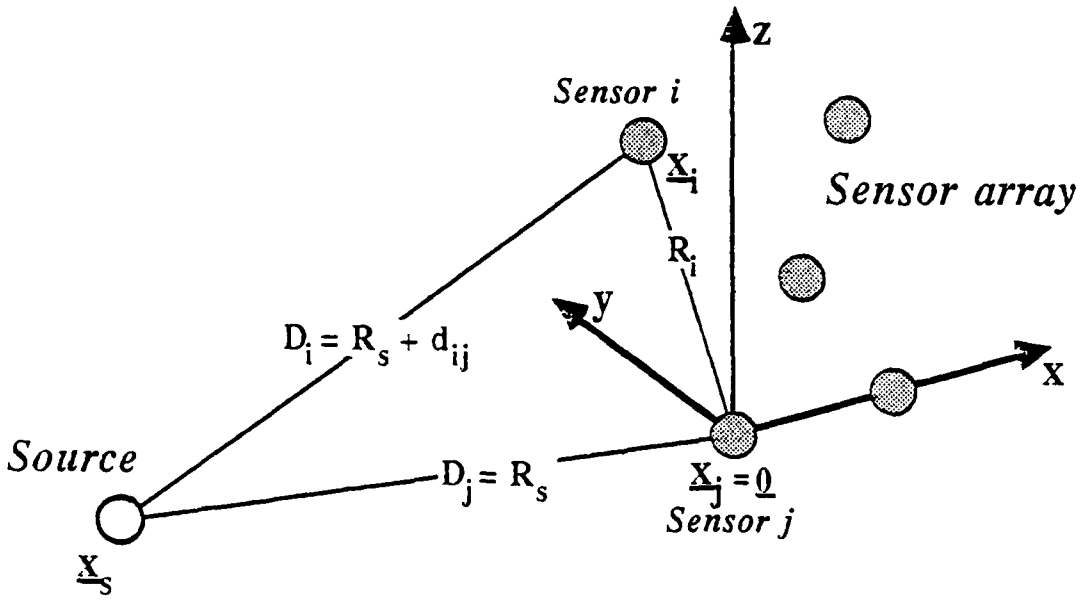

z
Sensor i
$\underline{x}_i$
Sensor array
R_i
y
$D_i = R_s + d_{ij}$
x
Source
$D_j = R_s$
$\underline{x}_j = \underline{0}$
Sensor j
$\underline{x}_s$

Figure 1b

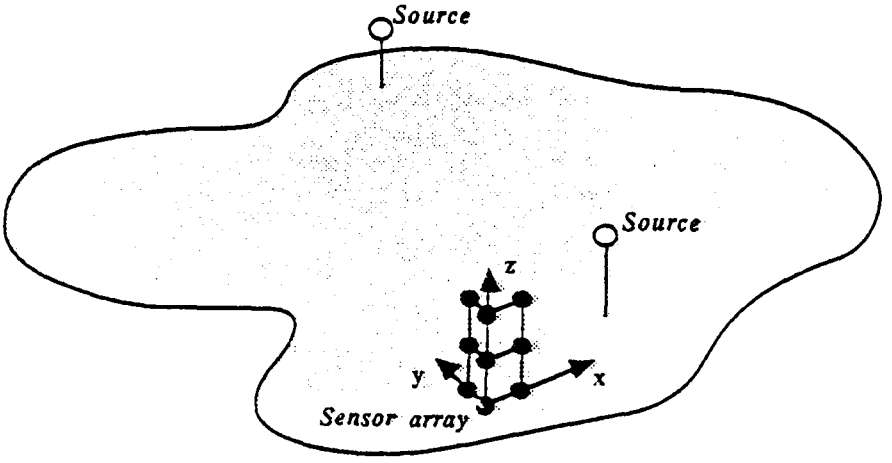


Figure 2

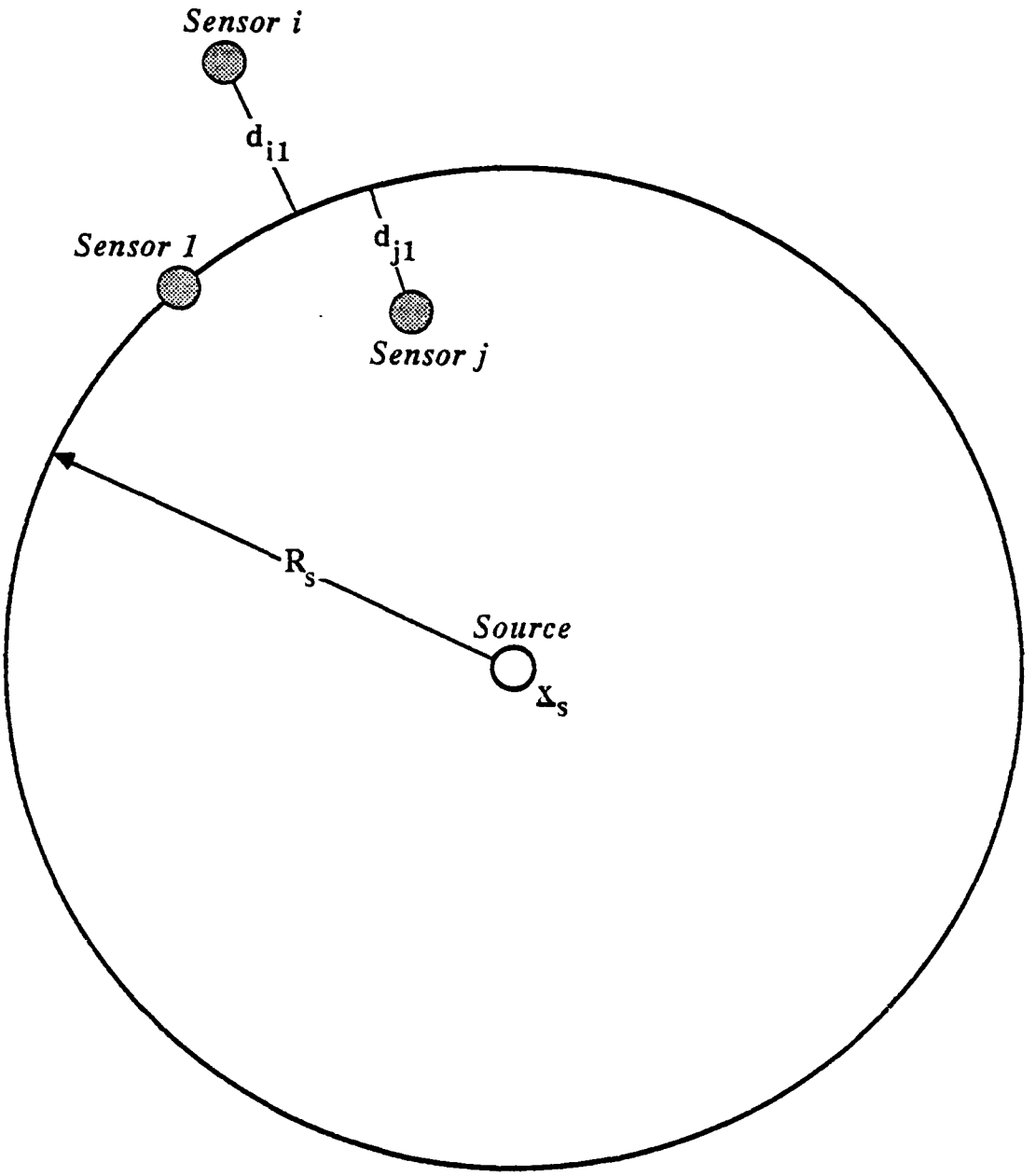


Figure 3

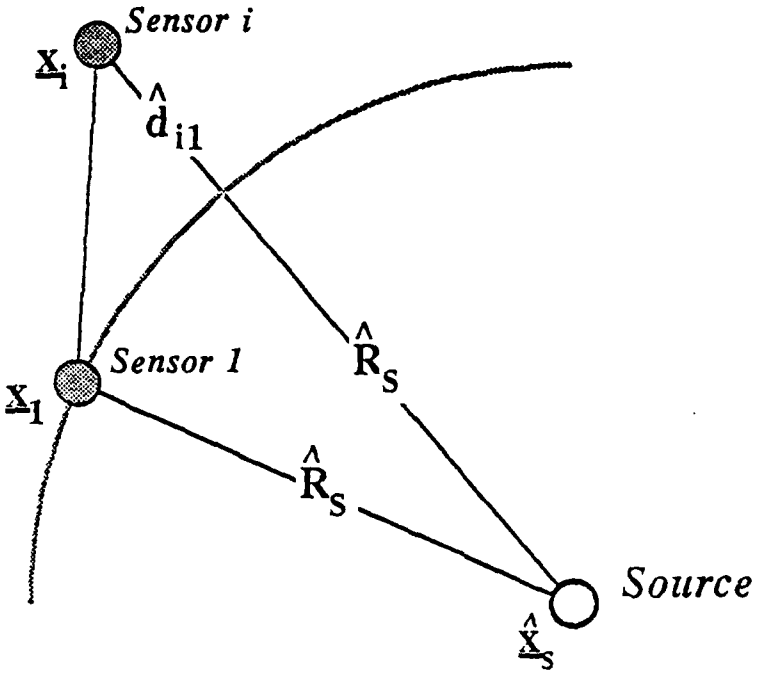

Sensor i
$\underline{x}_i$
$\hat{d}_{i1}$
Sensor 1
$\underline{x}_1$
$\hat{R}_S$
$\hat{R}_S$
Source
$\hat{\underline{x}}_S$

Figure 4

Appendix E - On the Efficiency of the Spherical Interpolation Estimator of Source Localization using Range Difference Measurements

On the Efficiency of the Spherical Interpolation Estimator of Source Location Using Range Difference Measurements

J. S. Abel and J. O. Smith

Systems Control Technology
1801 Page Mill Road
Palo Alto, California 94303

Abstract. Cramer-Rao lower bounds are derived for the variance of unbiased estimators of source location, source range, and source direction given range-difference (RD) measurements. The Cramer-Rao bound (CRB) for range estimate variance is found to be a quartic function of source range. The CRB for bearing estimate variance is found to be insensitive to the source location, an inverse quadratic function of the sensor-array solid-angle as seen from the source, and an inverse linear function of the spatial density of the sensors within the array (assuming independent RD measurements in each pair of sensors). The theoretical bounds are compared to empirical mean-squared errors obtained using a fast, suboptimal, least squares estimator of source location called the Spherical Interpolation (SI) method [1]. The SI method estimates' mean-squared errors are found to be in the range of 1.0 to 1.5 times the respective Cramer-Rao lower bounds.

1. Introduction

The problem of locating an object by analyzing signals received from it is a basic one in the fields of underwater acoustics, geophysics, aerospace and industry. It is therefore of interest to know how well a source localization algorithm can perform when the received signals are corrupted by noise.

A source tells of its location through the relative time delays measured among the signals it radiates to the sensors of an array. In a constant-velocity medium, the *time-difference-of-arrival* (TDOA) between signals received at two sensors is proportional to the difference in source-sensor range, termed *range difference* (RD), and places the source on a hyperboloid of revolution about a line drawn between the sensors. In p dimensions, RD measurements from p non-coplanar sensor pairs ($p+1$ sensors) are sufficient to localize a source [2].

Typically, the RD's are not known precisely and the source location has to be estimated. The source location estimate will have, under suitable assumptions, an associated variance and bias. The Cramer-Rao bound (CRB) gives a lower bound on the variance of an unbiased estimator, and is often used in evaluating the performance of an estimator. An unbiased estimator which achieves the CRB is called *efficient;* it is well known that all *Maximum Likelihood* (ML) estimators are efficient. Unfortunately, in the case of estimating source location from noisy RD's, the ML estimator is expensive to implement, requiring nonlinear optimization techniques. While much is known about the problem of estimating RD's from received signals (a recent special issue on the topic is [13]), there has been relatively little work on the estimation of source location from RD measurements [1,2,3,4,5].

This paper presents Cramer-Rao lower bounds on the variance of unbiased estimates of source location, direction, and range from RD measurements. These bounds are then compared to the measured variances of estimates obtained using the *Spherical Interpolation* (SI) method [1]. The SI method provides a fast, closed-form, least-squares type solution for source location from RD measurements, and is shown here to have RMS error approaching the Cramer-Rao lower bound.

The organization is as follows. In section 2 the Spherical-Interpolation Method of source location estimation is derived. In section 3 the Cramer-Rao lower bounds on the variance of source location, bearing, and range estimates are derived and discussed. Next, section 4 presents Monte-Carlo simulation results demonstrating the near-efficiency of the SI technique.

2. The Spherical-Interpolation Method for Closed-Form Localization

Let N denote the number of sensors, and let d_{ij} denote the RD between sensors i and j ($i,j = 1,\ldots,N$). The vector of (x, y, z) spatial coordinates for the ith sensor is denoted $\underline{x}_i$, and the position of the source is

denoted $\underline{x}_i$. The distance between the source and sensor i is denoted by $D_i = \|\underline{x}_i - \underline{x}_s\|$, and the distance from the origin to the point $\underline{x}_i$ is denoted R_i. Similarly, $R_s = \|\underline{x}_s\|$. These quantities appear in Figure 1a. Note that

$$d_{ij} \triangleq D_i - D_j, \qquad i = 1, \ldots, N, \quad j = 1, \ldots, N \tag{1}$$

The localization problem is to determine $\underline{x}_s$ given d_{ij} for i and j between 1 and N. Note that there are $N(N-1)/2$ distinct RD's d_{ij} (excluding $i = j$, and counting each $d_{ij} \equiv -d_{ji}$ pair once); however, any $N-1$ RD measurements which form a "minimal spanning subtree" determine all the rest (in the noiseless case). The redundancy of the complete set of RD measurements is used to increase noise immunity.

2.1. Equation-Error Formulation

To solve the localization problem, we first map the spatial origin to an arbitrary sensor, say the jth, i.e., $\underline{x}_j = \underline{0}$. The RD d_{ij} then reduces to

$$d_{ij} = \|\underline{x}_i - \underline{x}_s\| - \|\underline{x}_s\|$$

and, from the definitions above,

$$d_{ij} = (R_i^2 + R_s^2 - 2\underline{x}_i^T \underline{x}_s)^{\frac{1}{2}} - R_s \tag{2}$$

Adding R_s to both sides of (2) and squaring gives

$$(d_{ij} + R_s)^2 = R_i^2 + R_s^2 - 2\underline{x}_i^T \underline{x}_s \tag{3}$$

Equation (3) is the Pythagorean theorem, illustrated in Figure 1b. Moving all terms to the right side of (3), the R_s^2 terms cancel, and we are left with

$$0 = R_i^2 - d_{ij}^2 - 2R_s d_{ij} - 2\underline{x}_i^T \underline{x}_s \tag{4}$$

The jth equation is degenerate so we have $N-1$ equations in three unknowns $\underline{x}_s$.

As the delays are not known precisely, we introduce a so-called "linear equation error" [8] into the right-hand-side of (4), and minimize it in a least squares sense to provide an estimate of the true solution. Without loss of generality, let $j = 1$. Then (4) becomes

$$\epsilon_i = R_i^2 - d_{i1}^2 - 2R_s d_{i1} - 2x_i^T x_s, \quad i = 2, 3, \ldots, N \tag{5}$$

where ϵ_i is the linear equation error to be minimized. The set of $N-1$ equations (5) can be written in matrix notation as

$$\underline{\epsilon} = \underline{\delta} - 2R_s \underline{d} - 2\mathbf{S}\underline{x}_s \tag{6}$$

where

$$\underline{\delta} \triangleq \begin{pmatrix} R_2^2 - d_{21}^2 \\ R_3^2 - d_{31}^2 \\ \vdots \\ R_N^2 - d_{N1}^2 \end{pmatrix}, \qquad \underline{d} \triangleq \begin{pmatrix} d_{21} \\ d_{31} \\ \vdots \\ d_{N1} \end{pmatrix}, \qquad \mathbf{S} \triangleq \begin{pmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{pmatrix} \tag{7}$$

It is worth noting that equation (6) is *linear in* $\underline{x}_s$ *given* R_s, and it is *linear in* R_s *given* $\underline{x}_s$. Error vectors which are linear in the unknowns yield closed-form least squares solutions.

The least squares solution for $\underline{x}_s$ given R_s is

$$\underline{x}_s = \frac{1}{2}\mathbf{S}_W^{\#}(\underline{\delta} - 2R_s \underline{d}) \tag{8}$$

where the *weighted linear equation error energy* $J(\underline{x}_s) \triangleq \underline{\epsilon}^T \mathbf{W} \underline{\epsilon}$ is minimized for

$$\mathbf{S}_W^{\#} \triangleq (\mathbf{S}^T \mathbf{W} \mathbf{S})^{-1} \mathbf{S}^T \mathbf{W} \tag{9}$$

and $\mathbf{W}$ is symmetric positive definite (or simply diagonal and positive). If $\mathbf{W} = \mathbf{I}$, $\mathbf{S}^{\dagger}_{W}$ becomes the pseudo-inverse of $\mathbf{S}$.

To obtain a true least-squares estimate, it is necessary to minimize $J(\underline{x}_s)$ with respect to $\underline{x}_s$ while allowing R_s to vary, maintaining the relation $R_s = \|\underline{x}_s\|$. This, unfortunately, is a nonlinear minimization problem, and serious difficulties can occur when $J(\underline{x}_s)$ is not a convex functional of $\underline{x}_s$. However, the nonlinearity can be eliminated as described in the next section.

2.2. The Spherical Interpolation Method

The basic idea of the SI solution is to substitute (8) into (6) and minimize the linear equation error again, this time with respect to R_s. This, surprisingly, yields a linear least squares problem for finding R_s, and the solution is quick and inexpensive. The technique is made possible by the fact that the formal least squares estimate of $\underline{x}_s$ given R_s in (6) is itself linear in R_s. When the minimizing R_s value is found in this new linear equation, the corresponding value of $\underline{x}_s$ (via (8)) is automatically a minimizer of the squared equation-error norm with respect to $\underline{x}_s$ given this R_s.

Rewriting the linear equation error defined in (6) to eliminate $\underline{x}_s$ by subsituting the value from (8) yields

$$\underline{\varepsilon} = \underline{\delta} - 2R_s\underline{d} - \mathbf{S}\mathbf{S}^{\dagger}_{W}(\underline{\delta} - 2R_s\underline{d}) = (\mathbf{I} - \mathbf{S}\mathbf{S}^{\dagger}_{W})(\underline{\delta} - 2R_s\underline{d})$$

where $\mathbf{I}$ is the $N-1$ by $N-1$ identity matrix. Now, $\underline{\varepsilon}$ is *linear* in the single unknown R_s. Define the $N-1$ by $N-1$ matrices

$$\begin{aligned} \mathbf{P}_{\mathcal{S}} &\triangleq \mathbf{S}\mathbf{S}^{\dagger}_{W} = \mathbf{S}(\mathbf{S}^{T}\mathbf{W}\mathbf{S})^{-1}\mathbf{S}^{T}\mathbf{W} \\ \mathbf{P}^{\perp}_{\mathcal{S}} &\triangleq \mathbf{I} - \mathbf{S}\mathbf{S}^{\dagger}_{W} = \mathbf{I} - \mathbf{P}_{\mathcal{S}} \end{aligned} \tag{10}$$

The rank of $\mathbf{P}_{\mathcal{S}}$ is at most 3 regardless of its order $N-1$, and $\mathbf{P}^{\perp}_{\mathcal{S}}$ has rank at least $(N-1)-3$. Also, $\mathbf{P}_{\mathcal{S}}$ and $\mathbf{P}^{\perp}_{\mathcal{S}}$ are *idempotent*, i.e., $\mathbf{P}^{2}_{\mathcal{S}} = \mathbf{P}_{\mathcal{S}}$ and $(\mathbf{P}^{\perp}_{\mathcal{S}})^{2} = \mathbf{P}^{\perp}_{\mathcal{S}}$. Idempotent matrices can be interpreted as *projection* operators. For example, when $\mathbf{W} = \mathbf{I}$, the operator $\mathbf{P}_{\mathcal{S}}$ projects an $(N-1)$-vector into the subspace spanned by the columns of $\mathbf{S}$, and the operator $\mathbf{P}^{\perp}_{\mathcal{S}}$ projects into the orthogonal complement of the subspace spanned by the columns of $\mathbf{S}$. Thus, for example, $\mathbf{P}_{\mathcal{S}}\underline{x}$ is orthogonal to $\mathbf{P}^{\perp}_{\mathcal{S}}\underline{y}$ for every pair of vectors $\underline{x}$ and $\underline{y}$.

In the nondegenerate, four-sensor case, $\mathbf{P}_{\mathcal{S}} = I_3$, and the error $\underline{\varepsilon}$ is zero. In the more general case of N sensors,

$$\underline{\varepsilon} = \mathbf{P}^{\perp}_{\mathcal{S}}(\underline{\delta} - 2R_s\underline{d})$$

so that

$$J'(R_s) \triangleq \|\underline{\varepsilon}\|^{2}_{W} = \underline{\varepsilon}^{T}\mathbf{W}\underline{\varepsilon} = (\underline{\delta} - 2R_s\underline{d})^{T}\mathbf{P}^{\perp}_{\mathcal{S}}\mathbf{W}\mathbf{P}^{\perp}_{\mathcal{S}}(\underline{\delta} - 2R_s\underline{d}) \triangleq (\underline{\delta} - 2R_s\underline{d})^{T}\mathbf{W}_{\mathcal{S}\perp}(\underline{\delta} - 2R_s\underline{d})$$

Minimizing J' with respect to R_s is a form of *weighted least squares* in which the weighting matrix $\mathbf{W}_{\mathcal{S}\perp}$ is of rank $N-4$. The missing dimensions reflect the degrees of freedom removed by choosing sensor 1 as the origin and substituting in the least-squares solution (8) for the three spatial source coordinates. Note that when $\mathbf{W} = \mathbf{I}$, $\mathbf{W}_{\mathcal{S}\perp} = \mathbf{P}^{\perp}_{\mathcal{S}}$. The least-squares minimizer of $J'(R_s)$ is given by

$$\tilde{R}_s = \frac{1}{2}\frac{\underline{d}^{T}\mathbf{W}_{\mathcal{S}\perp}\underline{\delta}}{\underline{d}^{T}\mathbf{W}_{\mathcal{S}\perp}\underline{d}} \tag{11}$$

Substituting this solution into (8) yields the closed-form source location estimate

$$\hat{\underline{x}}_{SI} = \frac{1}{2}\mathbf{S}^{\dagger}_{W}\left(\mathbf{I} - \frac{\underline{d}\,\underline{d}^{T}\mathbf{W}_{\mathcal{S}\perp}}{\underline{d}^{T}\mathbf{W}_{\mathcal{S}\perp}\underline{d}}\right)\underline{\delta} = \frac{1}{2}\mathbf{S}^{\dagger}_{W}(\underline{\delta} - 2\tilde{R}_s\underline{d}) \tag{12}$$

Clearly, the computational burden of (12) is very low compared to iterative nonlinear minimization.

Note that, in general, $\tilde{R}_s$ as computed by (11) is not necessarily equal to $\|\hat{\underline{x}}_{SI}\|$ computed from (12), except as the SD noise approaches zero. Therefore, we define the *range estimate* by

$$\hat{R}_{SI} \triangleq \|\hat{\underline{x}}_{SI}\| \tag{13}$$

Similarly, the *bearing estimate* is defined as the vector of direction cosines from the origin at sensor 1 to the source:

$$\hat{\underline{\Omega}}_{s1} \triangleq \frac{\hat{\underline{x}}_{s1}}{\| \hat{\underline{x}}_{s1} \|} \tag{14}$$

3. The Cramer-Rao Lower Bound

The Cramer-Rao bound (CRB) is a lower bound on the variance of an unbiased estimator [9,10,11,12]; estimators achieving this bound are said to be *efficient*. In this section CRB's are derived for unbiased estimators of source location, source range, and source direction based on RD information.

We first review the CRB, the reparameterization relation, and the information inequality. The variance of estimating the ith element of $\underline{\theta}$ with an unbiased estimator $\hat{\theta}_i$ based on an observation vector $\underline{x}$ is bounded below by the ith diagonal element of the inverse of the Fisher information matrix [9]:

$$\mathrm{Var}(\hat{\theta}_i) \triangleq \mathrm{E}\left\{\hat{\theta}_i^2\right\} - \mathrm{E}^2\left\{\hat{\theta}_i\right\} \geq \left[\underline{\underline{\mathcal{F}}}_{\underline{\theta}}^{-1}\right]_{ii} \tag{15}$$

where $\mathrm{Var}(\cdot)$ denotes variance, $\mathrm{E}\{\cdot\}$ denotes expectation, $[\mathbf{M}]_{ii}$ denotes the ith diagonal element of the matrix $\mathbf{M}$, and $\underline{\underline{\mathcal{F}}}_{\underline{\theta}}$ is the Fisher information matrix for $\underline{\theta}$, defined by [9]

$$\underline{\underline{\mathcal{F}}}_{\underline{\theta}} \triangleq \mathrm{E}\left\{\left[\frac{\partial}{\partial \underline{\theta}} \ln p(\underline{x}\,|\,\underline{\theta})\right]^{\mathrm{T}}\left[\frac{\partial}{\partial \underline{\theta}} \ln p(\underline{x}\,|\,\underline{\theta})\right]\right\}$$

If $\underline{\theta}$ is parametrized by t (i.e., $\underline{\theta} = h(\underline{t})$), the Fisher information matrix for t can be written in terms of the Fisher information matrix for $\underline{\theta}$ via the reparameterization relation [9]

$$\underline{\underline{\mathcal{F}}}_{\underline{t}} = \left(\frac{\partial \underline{\theta}}{\partial \underline{t}}\right)^{\mathrm{T}} \underline{\underline{\mathcal{F}}}_{\underline{\theta}} \left(\frac{\partial \underline{\theta}}{\partial \underline{t}}\right) \tag{16}$$

where, the (i,j)th element of $\partial\underline{\theta}/\partial\underline{t}$ is $\partial\theta_i/\partial t_j$. Finally, if $\hat{\gamma}$ is an unbiased estimator of the scalar $\gamma(\underline{\theta})$, the variance of $\hat{\gamma}$ is bounded by the information inequality:

$$\mathrm{Var}(\hat{\gamma}) \geq \left(\frac{\partial \gamma}{\partial \underline{\theta}}\right)^{\mathrm{T}} \underline{\underline{\mathcal{F}}}_{\underline{\theta}}^{-1} \left(\frac{\partial \gamma}{\partial \underline{\theta}}\right) \tag{17}$$

We now derive the CRB's for source location, range and direction. Defining $\hat{\underline{x}}_s(i)$ as the ith element of $\hat{\underline{x}}_s$, the Cramer-Rao lower bound on the variance of an unbiased estimator of source location $\underline{x}_s$, based on RD estimates $\hat{\underline{d}}$, is given by (15) as

$$\mathrm{Var}(\hat{\underline{x}}_s(i)) \geq \left[\underline{\underline{\mathcal{F}}}_{\underline{x}_s}^{-1}\right]_{ii} \tag{18}$$

where, $\underline{\underline{\mathcal{F}}}_{\underline{x}_s}$ is the Fisher information matrix for $\underline{x}_s$. Using the information inequality (17), the variances in estimating source range and source direction from RD measurements are bounded below by

$$\begin{aligned} \mathrm{Var}(\hat{R}_s) &\geq \left(\frac{\partial R_s}{\partial \underline{x}_s}\right)^{\mathrm{T}} \underline{\underline{\mathcal{F}}}_{\underline{x}_s}^{-1} \left(\frac{\partial R_s}{\partial \underline{x}_s}\right) \\ \mathrm{Var}(\hat{\underline{\Omega}}_s(i)) &\geq \left[\left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)^{\mathrm{T}} \underline{\underline{\mathcal{F}}}_{\underline{x}_s}^{-1} \left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)\right]_{ii} \end{aligned} \tag{19}$$

where $\hat{\underline{\Omega}}_s(i)$ is the ith element of $\hat{\underline{\Omega}}_s$.

Since the source location $\underline{x}_s$ is a parameter in determining the distribution of the RD measurement vector $\underline{d}$, by the reparameterization relation (16), $\underline{\underline{\mathcal{F}}}_{\underline{x}_s}$ can be written as

$$\underline{\underline{\mathcal{F}}}_{\underline{x}_s} = \left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right)^{\mathrm{T}} \underline{\underline{\mathcal{F}}}_{\underline{d}} \left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right) \tag{20}$$

and, as shown in Appendix A, assuming the RD estimates are Gaussian distributed with correlation matrix $R_{\underline{d}}$, assumed independent of source location, $\Gamma_{\underline{d}} = R_{\underline{d}}^{-1}$ and

$$\Gamma_{\underline{x}_s} = \left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right)^T R_{\underline{d}}^{-1} \left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right) \tag{21}$$

It should be noted that when the RD's are estimated from Gaussian signals, commonly used estimators are unbiased and have [6,7]

$$R_{\underline{d}} \approx \sigma_{\underline{d}}^2 \mathbf{I} \tag{22}$$

where $\sigma_{\underline{d}}^2$ is the variance of each individual RD and is dependent only on signal and medium characteristics.

To find the Cramer-Rao lower bound on the variance of the source location estimate, it remains only to find the sensitivity (derivative) of range difference with respect to source location $\partial \underline{d} / \partial \underline{x}_s$. Similarly, the CRB's for range and bearing of the source are easily calculated after finding the sensitivities of range and bearing with respect to source location.

3.1. Sensitivity of Range Difference to Source Location

The derivative $\partial \underline{d} / \partial \underline{x}_s$ can be evaluated by differentiating equation (6) in the noise-free case with respect to $\underline{x}_s$ and solving for $\partial \underline{d} / \partial \underline{x}_s$. Setting $\underline{\epsilon} = \underline{0}$ in (6) and differentiating with respect to $\underline{x}_s$ yields

$$\frac{\partial \underline{\delta}}{\partial \underline{x}_s} - 2\frac{\partial (R_s \underline{d})}{\partial \underline{x}_s} - 2\mathbf{S}\frac{\partial \underline{x}_s}{\partial \underline{x}_s} = \mathbf{0} \tag{23}$$

From (7), the ith element of $\underline{\delta}$ is $R_i^2 - d_{i1}^2$ so that $\partial \underline{\delta} / \partial \underline{x}_s$ becomes

$$\frac{\partial \underline{\delta}}{\partial \underline{d}} = \left(\frac{\partial \underline{\delta}}{\partial \underline{d}}\right)\left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right) = -2\boldsymbol{\Delta}_{\underline{d}} \frac{\partial \underline{d}}{\partial \underline{x}_s} \tag{24}$$

where, $\boldsymbol{\Delta}_{\underline{d}}$ is a diagonal matrix with $\underline{d}_i = d_{i1}$ as the ith row/column entry. The derivative, $\partial (R_s \underline{d}) / \partial \underline{x}_s$, is evaluated using the product rule:

$$\frac{\partial (R_s \underline{d})}{\partial \underline{x}_s} = R_s \frac{\partial \underline{d}}{\partial \underline{x}_s} + \underline{d}\frac{\partial R_s}{\partial \underline{x}_s} = R_s \frac{\partial \underline{d}}{\partial \underline{x}_s} + \underline{d}\,\underline{\Omega}_s^T \tag{25}$$

where, $R_s = \| \underline{x}_s \|$ has been used to evaluate $\partial R_s / \partial x_s = \Omega_s \triangleq x_s / \| \underline{x}_s \|$. Substituting (24) and (25) into (23) with $\partial \underline{x}_s / \partial \underline{x}_s = \mathbf{I}$ yields $\partial \underline{d} / \partial \underline{x}_s$:

$$\boldsymbol{\Delta}\frac{\partial \underline{d}}{\partial \underline{x}_s} + \boldsymbol{\Delta} = \mathbf{0} \;\Rightarrow\; \frac{\partial \underline{d}}{\partial \underline{x}_s} = -\mathbf{V}\boldsymbol{\Delta} \tag{26}$$

where

$$\begin{aligned} \mathbf{V} &\triangleq \boldsymbol{\Delta}^{-1} \\ \boldsymbol{\Delta} &\triangleq \boldsymbol{\Delta}_{\underline{d}} - R_s \mathbf{I} \\ \boldsymbol{\Delta} &\triangleq \underline{d}\,\underline{\Omega}_s^T - \mathbf{S} \end{aligned} \tag{27}$$

Post-multiplying through (6) by $\underline{\Omega}_s^T / R_s$, solving for $\underline{d}\,\underline{\Omega}_s^T$, and substituting into (27) produces a formula for $\boldsymbol{\Delta}$ in terms of $\underline{\delta}$ rather than $\underline{d}$:

$$\boldsymbol{\Delta} = \mathbf{S}\mathbf{P}_{\underline{x}_s}^{\perp} - \frac{\underline{\delta}\,\underline{\Omega}_s^T}{2R_s}\mathbf{P}_{\underline{x}_s} \tag{28}$$

where

$$\begin{aligned} \mathbf{P}_{\underline{x}_s} &\triangleq \underline{x}_s (\underline{x}_s^T \underline{x}_s)^{-1} \underline{x}_s^T = \underline{\Omega}_s \underline{\Omega}_s^T \\ \mathbf{P}_{\underline{x}_s}^{\perp} &\triangleq (\mathbf{I} - \mathbf{P}_{\underline{x}_s}) \end{aligned}$$

The idempotent matricies $\mathbf{P}_{\underline{x}_s}$ and $\mathbf{P}_{\underline{x}_s}^{\perp}$ are projection operators which remove the vector components perpendicular to $\underline{x}_s$ and parallel to $\underline{x}_s$, respectively. Thus, $\boldsymbol{\Delta}$ multiplies the 2D subspace orthogonal to $\underline{\Omega}_s$ by the matrix $\mathbf{S}$, and it "bends" the 1D component parallel to $\underline{\Omega}_s$ into the direction of $\underline{\delta}$, scaled by $\| \underline{\delta} \| / 2R_s$.

3.2. Cramer-Rao Bounds for Location, Range, and Bearing

The Cramer-Rao lower bound on the variance of an estimator of source *location*, $\hat{\underline{x}}_s$, from Gaussian distributed RD measurements is obtained by using (26) in (21):

$$\mathrm{Var}(\hat{\underline{x}}_s(i)) \geq \left[\mathcal{F}_{\underline{x}_s}^{-1}\right]_{ii} \tag{29}$$

where

$$\begin{aligned}\mathcal{F}_{\underline{x}_s}^{-1} &= \left[\left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right)^T R_{\underline{d}}^{-1}\left(\frac{\partial \underline{d}}{\partial \underline{x}_s}\right)\right]^{-1} \\ &= \left(\boldsymbol{\Delta}^T \mathbf{V} R_{\underline{d}}^{-1} \mathbf{V} \boldsymbol{\Delta}\right)^{-1} \\ &\approx \sigma_{\underline{d}}^2 \left(\boldsymbol{\Delta}^T \mathbf{V}^2 \boldsymbol{\Delta}\right)^{-1}\end{aligned} \tag{30}$$

The Cramer-Rao lower bound on the variance in estimating *source range*, R_s, from RD measurements is similarly

$$\begin{aligned}\mathrm{Var}(\hat{R}_s) \geq \left(\frac{\partial R_s}{\partial \underline{x}_s}\right)\mathcal{F}_{\underline{x}_s}^{-1}\left(\frac{\partial R_s}{\partial \underline{x}_s}\right)^T &= \underline{\Omega}_s^T \mathcal{F}_{\underline{x}_s}^{-1} \underline{\Omega}_s \\ &= \underline{\Omega}_s^T \left(\boldsymbol{\Delta}^T \mathbf{V} R_{\underline{d}}^{-1} \mathbf{V} \boldsymbol{\Delta}\right)^{-1} \underline{\Omega}_s \\ &\approx \sigma_{\underline{d}}^2 \underline{\Omega}_s^T \left(\boldsymbol{\Delta}^T \mathbf{V}^2 \boldsymbol{\Delta}\right)^{-1} \underline{\Omega}_s\end{aligned} \tag{31}$$

For the bearing estimate CRB, we need (using $\underline{\Omega}_s \triangleq \underline{x}_s / \|\underline{x}_s\|$):

$$\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s} = (\underline{x}_s^T \underline{x}_s)^{-\frac{1}{2}}\left[\mathbf{I} - \frac{\underline{x}_s \underline{x}_s^T}{(\underline{x}_s^T \underline{x}_s)}\right] = \frac{1}{R_s}(\mathbf{I} - \underline{\Omega}_s \underline{\Omega}_s^T) \triangleq \frac{\mathbf{P}_{\underline{x}_s}^{\perp}}{R_s} \tag{32}$$

The CRB for the *bearing estimate*, $\hat{\underline{\Omega}}_s$, is thus

$$\mathrm{Var}(\hat{\underline{\Omega}}_s(i)) \geq \left[\left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)\mathcal{F}_{\underline{x}_s}^{-1}\left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)^T\right]_{ii} \tag{33}$$

where

$$\begin{aligned}\left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)\mathcal{F}_{\underline{x}_s}^{-1}\left(\frac{\partial \underline{\Omega}_s}{\partial \underline{x}_s}\right)^T &= \frac{1}{R_s^2}\mathbf{P}_{\underline{x}_s}^{\perp T}\mathcal{F}_{\underline{x}_s}^{-1}\mathbf{P}_{\underline{x}_s}^{\perp} \\ &= \frac{1}{R_s^2}\mathbf{P}_{\underline{x}_s}^{\perp}\left(\boldsymbol{\Delta}^T \mathbf{V} R_{\underline{d}}^{-1} \mathbf{V} \boldsymbol{\Delta}\right)^{-1}\mathbf{P}_{\underline{x}_s}^{\perp} \\ &\approx \frac{\sigma_{\underline{d}}^2}{R_s^2}\mathbf{P}_{\underline{x}_s}^{\perp}\left(\boldsymbol{\Delta}^T \mathbf{V}^2 \boldsymbol{\Delta}\right)^{-1}\mathbf{P}_{\underline{x}_s}^{\perp}\end{aligned} \tag{34}$$

3.3. Far-Field Analysis

In this section, we analyze the location, range, and bearing CRB's in the "far field" ($R_s \gg d_{11}, \forall i$) and the "infinite-range case" ($R_s \to \infty$). These assumptions on the scale of R_s greatly simplify the CRB expressions and give insight into the behavior of the bounds. We only consider the i.i.d. case $R_{\underline{d}} = \sigma_{\underline{d}}^2 \mathbf{I}$, as is often done in practice [6,7].

The far field

The far field of the sensor-array aperture is defined by

$$R_s \gg d_{i1}, \quad i = 2, \ldots, N \tag{35}$$

At this range, we have $\Delta \triangleq R_s \mathbf{I} - \Delta_{\underline{d}} \approx R_s \mathbf{I}$, and from (22), (27), and (29)

$$\mathcal{F}_{\underline{x}_s}^{-1} \approx \sigma_{\underline{d}}^2 R_s^2 (\Delta^T \Delta)^{-1} \tag{36}$$

Using (28) for Δ and (31) for the CRB variance of R_s, (36) gives the far-field range-variance bound as

$$\mathrm{Var}(R_s) \gtrsim 4\sigma_{\underline{d}}^2 R_s^4 \left[\frac{1}{\underline{\delta}^T \mathbf{P}_{\tilde{\mathbf{U}}} \underline{\delta}} \right] \tag{37}$$

where

$$\begin{aligned} \mathbf{P}_{\tilde{\mathbf{U}}} &= \mathbf{I} - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T \\ \mathbf{U} &= \mathbf{S}\Omega_s^\perp \\ \Omega_s^\perp &= \left[\underline{\Omega}_1^\perp \quad \underline{\Omega}_2^\perp \right] \end{aligned} \tag{38}$$

where $\underline{\Omega}_1^\perp$ and $\underline{\Omega}_2^\perp$ are any two mutually orthogonal vectors in 3-space which are also orthogonal to $\underline{\Omega}_s$. The matrix $\mathbf{P}_{\tilde{\mathbf{U}}}$ is the idempotent projection into the orthogonal complement of the subspace spanned by the columns of $\mathbf{U} = \mathbf{S}\Omega_s^\perp$.

Similarly, the far-field bearing-variance bound is approximated by

$$\begin{aligned} \mathrm{Var}(\hat{\Omega}_s(i)) &\gtrsim \left[\sigma_{\underline{d}}^2 \Omega_s^\perp \left(\mathbf{U}^T \mathbf{P}_{\underline{\delta}^\perp} \mathbf{U} \right)^{-1} \Omega_s^{\perp T} \right]_{ii} \\ \mathbf{P}_{\underline{\delta}^\perp} &= \mathbf{I} - \frac{\underline{\delta}\,\underline{\delta}^T}{\underline{\delta}^T \underline{\delta}} \end{aligned} \tag{39}$$

and again $\mathbf{U} = \mathbf{S}\Omega_s^\perp$.

We now discuss qualitative relationships revealed by the above formulas. Equations (37) and (39) show that the far-field *range estimate* Cramer-Rao lower bound variance increases quartically with the range, and the far-field *bearing estimate* deviation is constant with source range. Consequently, the minimum standard deviation of a *source location* estimator increases with the square of the source range in the source direction, and increases linearly with the range in the other directions.

Second from the far-field bounds, it is apparent that increasing the norm of $\Delta^T\Delta$ reduces the Cramer-Rao bound variance in estimating source location, direction, and range. By the definition of Δ given in (28), therefore, the far-field bounds are reduced when the size of $\mathbf{S}$ in the directions perpendicular to the source direction increases, or when the norm of $\underline{\delta}$ increases. The sizes of $\mathbf{S}$ and $\underline{\delta}$ are increased by placing the sensors far from each other and far from sensor 1, as seen by the source, i.e. by increasing the size of the array. The norm of $\underline{\delta}$ is also increased by aligning the array such that lines drawn between sensor 1 and the other sensors tend to be perpendicular to the source, thereby decreasing $\underline{d}$ and increasing $\underline{\delta}$. Increasing $\mathbf{S}$ or $\underline{\delta}$ lowers the source estimate variance by increasing the effective aperture size of the sensor array, as seen from the source.

Third, the variances in the source location, direction, and range estimates are all proportional to the RD noise variance.

To corroborate the above observations, the exact Cramer-Rao bounds were evaluated numerically and compared to the far-field observations. Figure 2a shows the square root of the CRB range estimate standard deviation plotted against range for the case of a 100-meter-wide, 200-meter-long 9-sensor prism-shaped array locating a source placed at various ranges along a line through the origin (see Monte-Carlo Run 1 in Table 1). *The range estimate standard deviation can be seen to increase very linearly with the square of the range, even at small ranges.* Figure 2b shows the CRB direction-cosines estimate standard deviation plotted against range for the case described above. *The bearing estimate standard deviation can be seen to be constant with increasing range, except at small ranges where there appears to be a point of maximum angular resolution, or best focus.*

The Infinite-Range Case

In the infinite-range case, $R_s \to \infty$, only the source direction, $\underline{\Omega}_s$ is of interest. At infinite-range equation (6) gives

$$-\mathbf{S}\underline{\Omega}_s = \underline{d} \;\Rightarrow\; \underline{\Omega}_s = -\mathbf{S}^{\bullet}\underline{d} \tag{40}$$

where $\mathbf{S}^{\bullet} \triangleq (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$. Note that by this defining relationsip, the infinite-range $\underline{\Omega}_s$ is not constrained to have unit norm.

The Fisher information matrix for the infinite-range $\underline{\Omega}_s$ estimate is easily computed from $\mathbf{F}_{\underline{d}} = R_{\underline{d}}^{-1}$ and the reparameterization relation; accordingly the infinite-range Cramer-Rao bound for the *bearing estimate* is

$$\mathrm{Var}(\hat{\Omega}_s(i)) \geq \left[\mathbf{F}_{\underline{\Omega}_s}^{-1}\right]_{ii} \tag{41}$$

where

$$\mathbf{F}_{\underline{\Omega}_s}^{-1} = \left[\left(\frac{\partial \underline{d}}{\partial \underline{\Omega}_s}\right)^T R_{\underline{d}}^{-1}\left(\frac{\partial \underline{d}}{\partial \underline{\Omega}_s}\right)\right]^{-1} = \left[\mathbf{S}^T R_{\underline{d}}^{-1}\mathbf{S}\right]^{-1} = \mathbf{S}^{\bullet T} R_{\underline{d}}\mathbf{S}^{\bullet} \approx \sigma_{\underline{d}}^2(\mathbf{S}^T\mathbf{S})^{-1}$$

Note that this bound is independent of source direction (since $\|\hat{\Omega}_s\|$ is unconsrtained). It is also inversely proportional to the square of the scale of the array, i.e., if the array were twice as large, the RD variance in estimating $\underline{\Omega}_s$ would be reduced by a factor of four. The bound is also inversely proportional to sensor-array density; e.g., if the number of sensors is doubled by measuring two independent RD estimates at each sensor (thus doubling the length of $\mathbf{S}$ by repeating each row), the value of $\mathbf{S}^T\mathbf{S}$ doubles and the variance of $\underline{x}_s$ halves. In practice, the bound is inversely proportional to array density up to the density limit for which the RD's can be assumed uncorrelated. As always, the TDOA variance is proportional to the RD variance.

It is shown in Appendix B that the infinite-range lower bound (41) is achieved by the ST method.

4. Monte-Carlo Simulation Results

This section presents Monte-Carlo simulation results comparing the performance of the ST method to the Cramer-Rao lower bound. The simulations were implemented in the Ctrl-C* language on a Vax 11/785 computer. Since the ST estimator exhibits a slight bias, the *root-mean-squared (RMS) error*, defined by,

$$\sigma_{\underline{x}_s} \triangleq \left[E(\hat{\underline{x}}_s - \underline{x}_s)^2\right]^{\frac{1}{2}}$$

of the ST method was compared to the Cramer-Rao lower bound standard deviation. It was found that the ST method is nearly efficient in all cases tried.

The simulations employed the two nine-sensor arrays shown in Figure 3, with the two source locations shown and two choices of sensor 1. The RD measurements were simulated by adding i.i.d. white Gaussian pseudo-random noise to the true RD values (corresponding to the practical assumption $R_{\underline{d}} = \sigma_{\underline{d}}^2\mathbf{I}$ [1,6,7]). Sample bias, variance, and RMS error of the unweighted ST method were computed by averaging the results of 100-trial Monte-Carlo runs. The Cramer-Rao bounds for each Monte-Carlo run were computed using the formulas given in section 3.2 above.

Table 1 describes the environmental information (source location, range, bearing, additive RD noise level, and sensor array) for each of the eight 100-trial Monte-Carlo runs.

Tables 2, 3 and 4 show the sample RMS error of the ST source location, range and bearing estimates and the corresponding Cramer-Rao bounds for each of the Monte-Carlo runs. It should be noted that the sample biases of the ST estimates, not shown here, were typically less than one tenth of the sample standard deviation, and were never prominent in determining the RMS error.

From Tables 2, 3, and 4, as well as other results not presented here, the ST estimators of range, bearing, and source location are seen to have RMS error in the range 1.0 to 1.5 times the Cramer-Rao lower bound standard deviation. The ST estimator is therefore performing almost efficiently. In other terms, the ST estimator approximates the maximum likelihood estimator of source location given RD measurements.

* Ctrl-C is a MatLab-based, high-level, "matrix calculator" language sold by SCT, Inc.

It is interesting to see that the choice of sensor 1 (the sensor to which the RD's are measured) has a large effect on the Cramer-Rao lower bound (as much as a factor of two in standard deviation). It has been observed that the range, and therefore source location estimates, have smaller CRB for sensor 1 choices which are either centrally located in the sensor array relative to the source or are closer to the source. Also observed is the tendency for the CRB to be smaller for source direction estimates for which sensor 1 is chosen to be centrally located in the sensor array as seen from the source.

The above observations are consistent with the far-field analysis of section 3. For these sensor 1 choices, R_s is smaller and $\| \underline{\dot{c}} \|$ is larger, both resulting in smaller Cramer-Rao lower bounds.

Also consistent with prior analysis is that the CRB's obtained using the larger (prism shaped) array, are lower than the corresponding bounds using the smaller (cube shaped) array. By the analysis of Section 3, when the array size increases, the matrix S is larger, and the CRB standard deviation is decreased.

The SI technique is seen to achieve the Cramer-Rao lower bound for some choices of sensor 1, while performing at as much as 1.5 times the CRB for other choices of sensor 1 for the same source location, RD noise, and sensor array. It has been observed, however, in all cases tried, that the choice of sensor 1 giving the lowest CRB also gives the lowest RMS error estimates using the SI technique. This suggests a three step procedure for locating a source: first, use an arbitrary sensor 1 to locate the source using the SI method; second, using this source location estimate, find the best choice for sensor 1 using the Cramer-Rao lower bound formulas above; and third, use the SI method to locate the source using RD measurements made from the best sensor 1. A final refinement would be to perform Gauss-Newton descent on $\| \underline{d} - \underline{\hat{d}}(\underline{x}_s) \|^2$ to converge to the optimal solution.

5. Summary

In this paper we presented Cramer-Rao bounds on the minimum variance in estimating source location, direction, and range from RD measurements. It was found that the CRB range estimate variance was a quartic function of the source range, and sensitive to the sensor array, and the choice of reference sensor. The source direction-cosines estimate variance was seen to be sensitive to the sensor array used, but insensitive to the source location.

The SI estimator of source location, direction, and range was described, and was shown to be asymptotically efficient (as the range approaches infinity) in the case of Gaussian distributed range-difference measurement. Additionally, the SI method was shown by computer simulation to a be in the range of 1.0 to 1.5 times the Cramer-Rao lower bound for a several near-field source locations.

Appendix A — Source Location Fisher Information Matrix for Gaussian Distributed Range-Difference Measurements

In this appendix, we derive the relationship (21).

Let $p(\hat{\underline{d}} \,|\, \underline{x}_s)$ denote the probability density function of $\hat{\underline{d}}$ given $\underline{x}_s$, where $\hat{d}$ is a random $(N-1)$-vector, and $\underline{x}_s$ is a vector of dimension 3 which parametrizes the distribution.

In the Gaussian case, we have [14]

$$p(\hat{\underline{d}} \,|\, \underline{x}_s) = \frac{1}{(2\pi)^{n/2} |R_{\underline{d}}|} e^{-\frac{1}{2}(\hat{\underline{d}}-\underline{d})^T R_{\underline{d}}^{-1}(\hat{\underline{d}}-\underline{d})}$$

where $\underline{d} = \mathrm{E}\{\hat{\underline{d}}\}$ is the mean (a function of $\underline{x}_s$) and $R_{\underline{d}} = \mathrm{E}\{\hat{\underline{d}}\hat{\underline{d}}^T\} - \underline{d}\,\underline{d}^T$ is the covariance matrix of the random vector $\hat{\underline{d}}$ (assumed independent of $\underline{x}_s$).

The Fisher information matrix is defined by [9]

$$\mathcal{F}_{\underline{x}_s} \triangleq \mathrm{E}\left\{ \left[\frac{\partial}{\partial \underline{x}_s} \ln p(\hat{\underline{d}} \,|\, \underline{x}_s) \right] \left[\frac{\partial}{\partial \underline{x}_s} \ln p(\hat{\underline{d}} \,|\, \underline{x}_s) \right]^T \right\}$$

The log of the distribution is

$$\ln p(\hat{\underline{d}} \,|\, \underline{x}_s) = -\frac{n}{2} \ln 2\pi - \ln |R_{\underline{d}}| - \frac{1}{2}(\hat{\underline{d}} - \underline{d})^T R_{\underline{d}}^{-1}(\hat{\underline{d}} - \underline{d})$$

so that

$$\frac{\partial \ln p(\hat{\underline{d}} \,|\, \underline{x}_s)}{\partial \underline{x}_s} = (\hat{\underline{d}} - \underline{d})^T R_{\underline{d}}^{-1} \left(\frac{\partial \underline{d}}{\partial \underline{x}_s} \right)$$

Thus,

$$\mathcal{F}_{\underline{x}_s} \triangleq \mathrm{E}\left\{ \left[\frac{\partial}{\partial \underline{x}_s} \ln p(\hat{\underline{d}} \,|\, \underline{x}_s) \right] \left[\frac{\partial}{\partial \underline{x}_s} \ln p(\hat{\underline{d}} \,|\, \underline{x}_s) \right]^T \right\} = \left(\frac{\partial \underline{d}}{\partial \underline{x}_s} \right)^T R_{\underline{d}}^{-1} \left(\frac{\partial \underline{d}}{\partial \underline{x}_s} \right)$$

The Cramer-Rao bound for an unbiased estimator of the parameter $\underline{x}_s$ based on $\hat{\underline{d}}$ observations is

$$\mathrm{Var}(\underline{x}_s) \geq \mathcal{F}_{\underline{x}_s}^{-1} = \left[\left(\frac{\partial \underline{d}}{\partial \underline{x}_s} \right)^T R_{\underline{d}}^{-1} \left(\frac{\partial \underline{d}}{\partial \underline{x}_s} \right) \right]^{-1}$$

Appendix B — Efficiency of the Infinite-Range SI Bearing Estimator

For an estimator to be efficient, it must be unbiased and have variance equal to the Cramer-Rao lower bound. Here, we show that the unweighted SI estimator of $\underline{\Omega}_s$ is efficient when the source is infinitely far away from the sensor array, assuming it is based on unbiased RD estimates $\hat{\underline{d}}$ with variance $R_{\underline{d}}$. (Efficiency of the bearing estimator is defined relative to the RD distribution. Therefore, the underlying RD estimator itself need not be efficient. It is also possible to allow the RD estimator to be biased as long as it is understood that in this case an unbiased bearing estimate is one whose mean is the bearing corresponding to the mean RD vector; in this case, the functional dependence of the biased RD mean on source bearing must be defined and differentiable. We prefer to assume an unbiased RD estimator since the added complexity of allowing RD bias does not appear to add significant practical value.)

We first show that the SI estimate $\hat{\underline{\Omega}}_{SI}$ of the infinite-range source direction-cosines vector. From (12), in the limit as $R_s \to \infty$,

$$\hat{\underline{\Omega}}_{SI} \triangleq \frac{\hat{\underline{x}}_s}{\hat{R}_s} = -\mathbf{S}^* \hat{\underline{d}}, \qquad R_s = \infty \tag{42}$$

where, $\mathbf{S}^* = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T$. Taking expected values,

$$E(\hat{\underline{\Omega}}_{SI}) = E(-\mathbf{S}^* \hat{\underline{d}}) = -\mathbf{S}^* E(\hat{\underline{d}}) = -\mathbf{S}^* \underline{d}$$

by the linearity of expectation and the fact that the RD estimate is unbiased.

The variance of the SI estimator is given by

$$\mathrm{Var}(\hat{\underline{\Omega}}_{SI}(i)) = \left[E(\hat{\underline{\Omega}}_{SI} \hat{\underline{\Omega}}_{SI}^T) - E(\hat{\underline{\Omega}}_{SI}) E(\hat{\underline{\Omega}}_{SI})^T \right]_{ii}$$

Substituting for $\hat{\underline{\Omega}}_{SI}$, we obtain

$$\begin{aligned} \mathrm{Var}(\hat{\underline{\Omega}}_{SI}(i)) &= \left[\mathbf{S}^* \left[E(\hat{\underline{d}} \hat{\underline{d}}^T) - E(\hat{\underline{d}}) E(\hat{\underline{d}})^T \right] \mathbf{S}^{*T} \right]_{ii} \\ &= \left[\mathbf{S}^* R_{\underline{d}} \mathbf{S}^{*T} \right]_{ii} \end{aligned}$$

which is the Cramer-Rao lower bound given in (41).

References

[1] J. O. Smith and J. S. Abel, "Closed-Form Least-Squares Localization of Multiple Broad-Band Emitters From Time-Difference-Of-Arrival Measurements," submitted for publication.

[2] J. P. Van Etten, "Navigation Systems: Fundamentals of Low and Very Low Frequency Hyperbolic Techniques," *Electrical Commun.*, vol. 45, no. 3, pp. 192–212, 1970.

[3] R. O. Schmidt, "A New Approach to Geometry of Range Difference Location," *IEEE Trans. Aero. and Elec. Systems*, vol. AES-8, no. 6, pp. 821–835, Nov. 1972.

[4] J. M. Delosme, M. Morf, and B. Friedlander, "A Linear Equation Approach to Locating Sources from Time-Difference-of-Arrival Measurements, " *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, 1980.

[5] H. C. Schau and A. Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," submitted for publication.

[6] B. Friedlander, "On the Cramer-Rao Bound for Time Delay and Doppler Estimation," *IEEE Trans. Info. Theory*, vol. IT-30, no. 3, pp. 575–580, May 1984.

[7] C. H. Knapp and G. C. Carter, "The Generalized Correleation Method for Estimation of Time Delay," *IEEE Trans. on Acoust. Speech, and Sig. Proc.*, vol. ASSP-23, No. 4, pp. 320–327, August 1976.

[8] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge MA, 1984.

[9] E. L. Lehmann, *Theory of Point Estimation*, John Wiley and Sons, New York, NY, pp. 123–129, 1983.

[10] R. A. Fisher, "On the Mathematical Foundations of Theoretical Statistics" *Phil. Trans. Roy. Soc., London*, **222**, 306, 1922.

[11] H. Cramer, *Mathematical Methods of Statistics*, Princeton University Press, Princeton, NJ, 1935.

[12] C. R. Rao, "Information and Accuracy Attainable in the Estimation of Statistical Parameters," *Bull. Calcutta Math. Soc.*, vol. 37, pp. 81–91, 1945.

[13] Special issue on time-delay estimation, *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, vol. ASSP-29, June 1981.

[14] A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics* McGraw-Hill, 1950, 1963, 1974.

Figure Captions.

Figure 1. Diagram illustrating notation and certain geometric relations for the case of a single source $\underline{x}_s$. Labels imbedded within a line denote the length of the corresponding vector. For example, $D_1 = \| \underline{x}_1 - \underline{x}_s \|$.

Figure 2a. The square root of the Cramer-Rao bound standard deviation for estimating source range plotted as a function of source range for the source direction, sensor array, and RD noise level given in Run 1, Table 1.

Figure 2b. The Cramer-Rao bound standard deviation for estimating source direction cosines plotted as a function of source range for the source direction, sensor array, and RD noise level used in generating Figure 2a.

Figure 3. Source-sensor geometry used in all simulations.

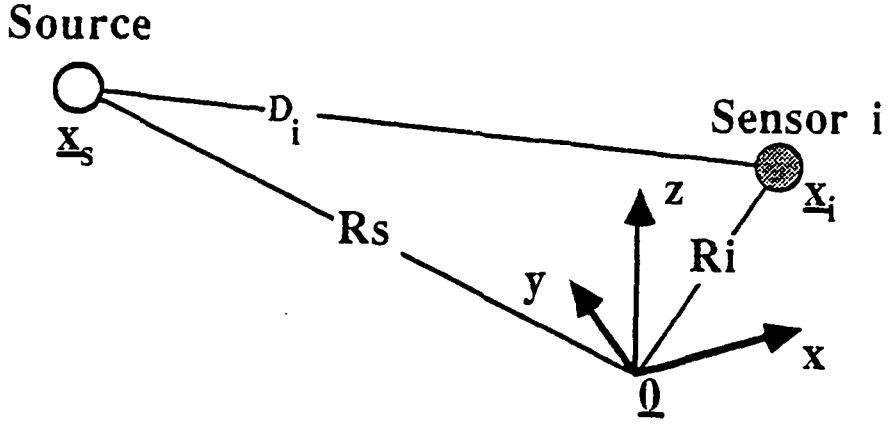


Figure 1a

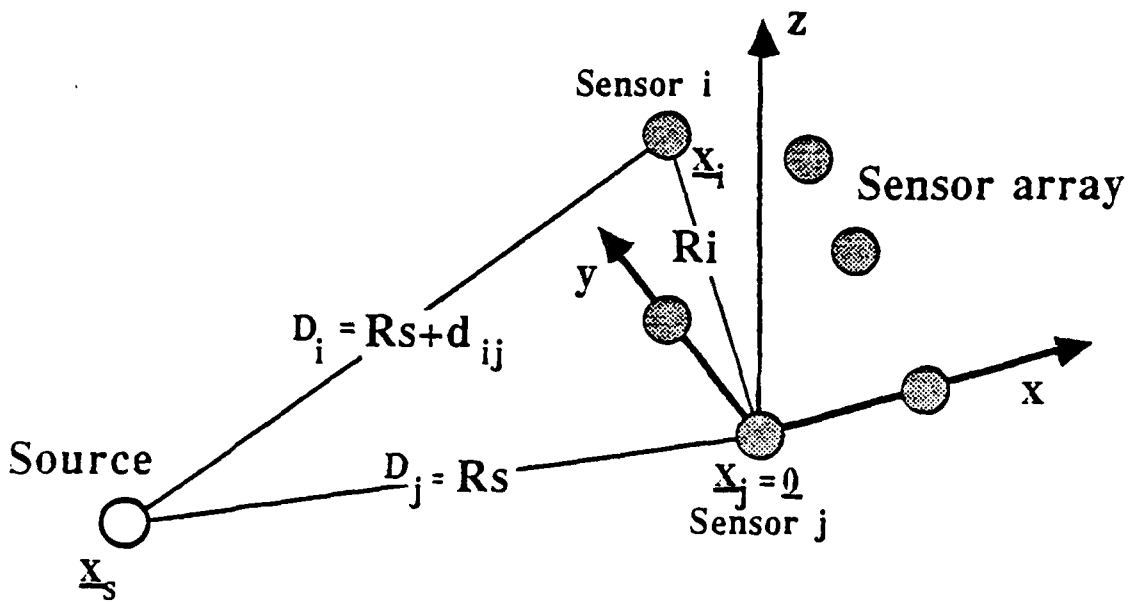

z
Sensor i
$\underline{x}_i$
Sensor array
Ri
y
$D_i = Rs+d_{ij}$
x
Source
$D_j = Rs$
$\underline{x}_j = \underline{0}$
Sensor j
$\underline{x}_S$

Figure 1b

Figure 2a

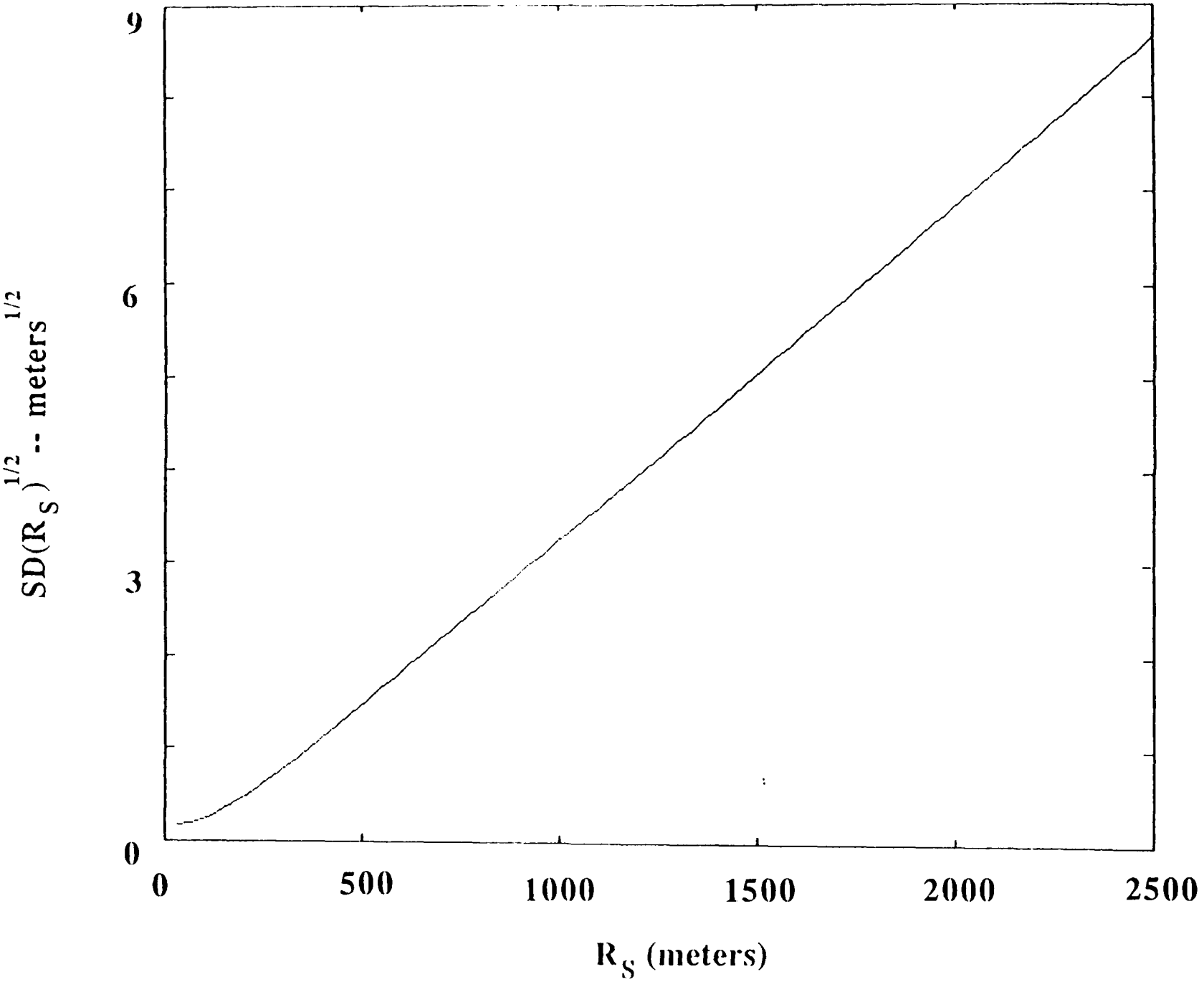
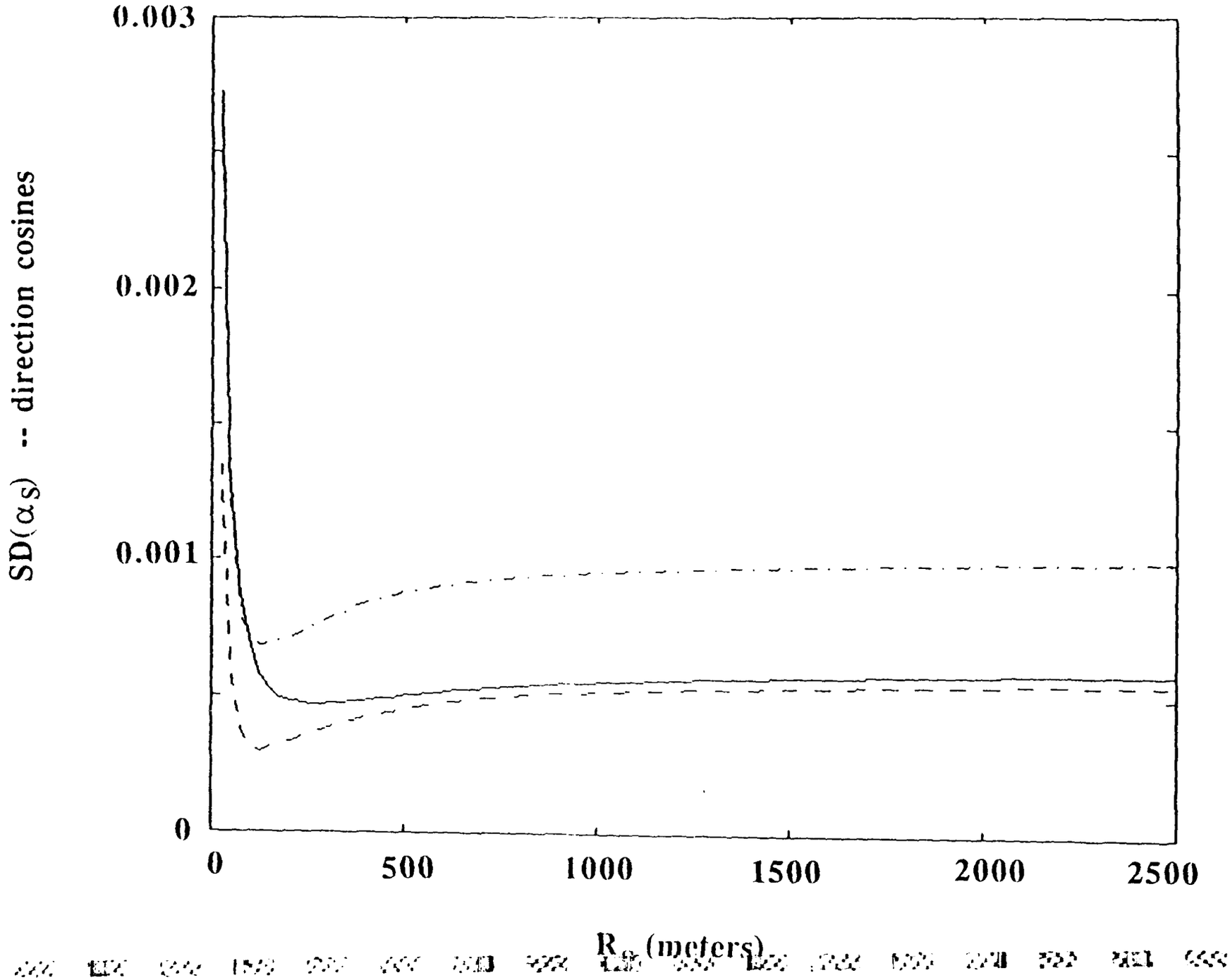


Figure 2b


0.003
0.002
0.001
0
SD(αs) -- direction cosines
0
500
1000
1500
2000
2500
Rs (meters)

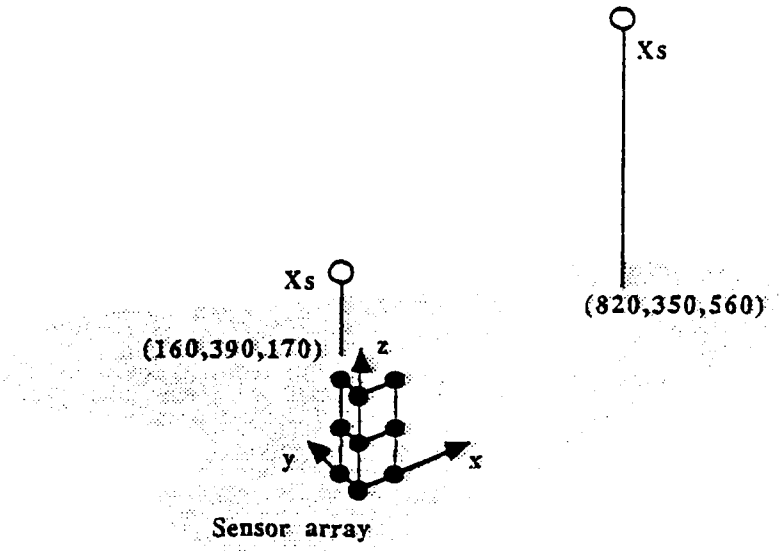


Figure 3a

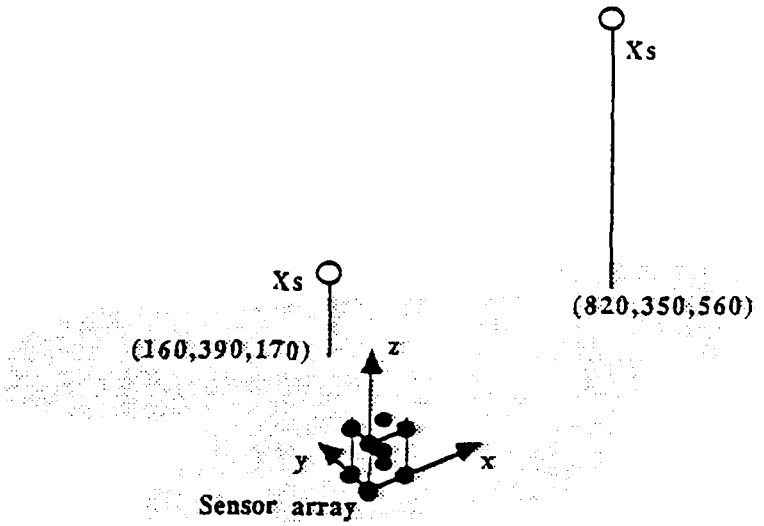


Figure 3b

Table 1
Monte-Carlo Runs

Run Number #	Sensor 1 (x,y,z) meters	Source Location (x,y,z) meters	Array #
1	(0,0,0)	(160,390,170)	1
2	(0,0,100)	(160,390,170)	1
3	(0,0,0)	(160,390,170)	2
4	(50,50,50)	(160,390,170)	2
5	(0,0,0)	(820,350,560)	1
6	(0,0,100)	(820,350,560)	1
7	(0,0,0)	(820,350,560)	2
8	(50,50,50)	(820,350,560)	2

Sensor Array 1:

0	0	0
0	0	100
0	0	200
0	100	0
0	100	100
0	100	200
100	0	0
100	0	100
100	0	200

Sensor Array 2:

0	0	0
0	0	100
0	100	0
100	0	0
0	100	100
100	100	0
100	0	100
100	100	100
50	50	50

σ_d = 1.0 meters, 100 trials per MC run

Table 2
SI Method Source Location RMS Error, Cramer-Rao Lower Bound

Run #	RMS Error, Cramer-Rao Lower Bound (meters)								
	SI σ_x	CRB σ_x	$\frac{\sigma_{x_{SI}}}{\sigma_{x_{CRB}}}$	SI σ_y	CRB σ_y	$\frac{\sigma_{y_{SI}}}{\sigma_{y_{CRB}}}$	SI σ_z	CRB σ_z	$\frac{\sigma_{z_{SI}}}{\sigma_{z_{CRB}}}$
1	6.0	6.2	0.98	18	17	1.07	3.9	3.0	1.30
2	3.7	3.4	1.07	14	9.4	1.50	3.6	2.4	1.51
3	11	7.1	1.54	32	23	1.39	11	7.8	1.48
4	6.3	6.1	1.03	16	17	0.99	6.6	6.6	1.01
5	108	91	1.18	41	36	1.16	64	51	1.24
6	91	62	1.48	33	23	1.44	60	39	1.53
7	147	116	1.27	56	43	1.31	97	76	1.28
8	94	94	1.00	37	37	0.99	62	63	0.99

Table 3
SI Method Source Direction Cosines RMS Error, Cramer-Rao Lower Bound

Run #	RMS Error, Cramer-Rao Lower Bound (direction cosine x 0.001)								
	SI σ_x	CRB σ_x	$\frac{\sigma_{x_{SI}}}{\sigma_{x_{CRB}}}$	SI σ_y	CRB σ_y	$\frac{\sigma_{y_{SI}}}{\sigma_{y_{CRB}}}$	SI σ_z	CRB σ_z	$\frac{\sigma_{z_{SI}}}{\sigma_{z_{CRB}}}$
1	7.5	5.0	1.51	6.0	4.4	1.34	9.4	8.6	1.09
2	9.8	5.5	1.78	6.6	2.3	2.91	4.1	3.8	1.07
3	8.1	6.7	1.21	5.6	4.3	1.29	7.6	6.5	1.18
4	8.0	6.9	1.17	8.2	3.1	2.66	6.9	6.7	1.02
5	6.8	5.8	1.18	8.1	5.4	1.50	9.4	7.5	1.25
6	7.8	2.9	2.73	1.1	6.4	1.70	4.7	4.3	1.10
7	6.2	4.9	1.27	9.4	7.5	1.25	6.5	5.6	1.15
8	7.8	4.3	1.81	8.0	6.7	1.19	6.7	6.0	1.11

Table 4
SI Method Source Range RMS Error, Cramer-Rao Lower Bound

Run	RMS Error, Cramer-Rao Lower Bound (meters)				
#	SI $\sigma_{\hat{R}_S}$	SI $\sigma_{\tilde{R}_S}$	CRB σ_{R_S}	$\frac{\sigma_{\hat{R}_S}}{\sigma_{R_S}}$	$\frac{\sigma_{\tilde{R}_S}}{\sigma_{R_S}}$
1	19	20	17	1.06	1.14
2	14	17	9.8	1.45	1.75
3	35	37	25	1.42	1.49
4	18	19	19	0.99	1.01
5	131	135	110	1.19	1.23
6	114	120	76	1.50	1.58
7	185	188	145	1.28	1.30
8	118	118	119	0.99	0.99

Appendix F - Track Parameter Estimation from Multipath Delay Information

TRACK PARAMETER ESTIMATION FROM MULTIPATH DELAY INFORMATION

J. S. Abel and Khosrow Lashkari

Systems Control Technology, Inc.
1801 Page Mill Road
Palo Alto, California 94304

Abstract

It is desired to track the location of an underwater acoustic source with range difference measurements from a stationary passive array. Many times, the array has only one or two sensors, and the multipath and intersensor range difference measurements are insufficient to localize and track a source moving along an arbitrary path [1]. Here, we propose to track sources with 1- or 2-sensor stationary passive arrays by making the simplifying assumption that the source's path can be described by a small set of so-called track parameters and using the range difference information to estimate the track parameter set rather than the source location as a function of time.

In this paper, we choose the track parameters to specify a straight-line, constant-velocity, constant-depth path. Cramer-Rao bounds are presented for estimating these track parameters from the time history of multipath and intersensor range difference measurements. It is shown that this track parameter set cannot be accurately estimated from the time history of a single multipath range difference without side information (an independent velocity estimate, for instance), although multipath and intersensor range difference

measurments from a two-sensor array are generally sufficient to estimate the track paremeter set. Computationally efficient techniques are presented which estimate track parameters from range difference measurements taken from 1- and 2-sensor arrays.

Monte-Carlo simulations are presented which show that these techniques have sample mean square error approximately equal to the Crammer-Rao bound when a single multipath range difference and an independent velocity estimate are available. The sample mean square error is shown to be in the range of two to ten times the corresponding Crammer-Rao bounds when these techniques are applied to 2-sensor range difference data.

1. INTRODUCTION

The tracking of a radiating source by a sensor array is a basic problem in underwater acoustics. In this paper, we describe methods for tracking sources using range difference measurements from passive stationary sensor arrays when the number of sensors is small and multipath surface reflections are present.

The signals recieved by the sensors of an array exhibit relative time delays corresponding to a sources' location. The relative time delay between two sensors, termed time difference of arrival (TDOA) is proportional to the source-sensor range difference (RD), and can be used to locate a source on a hyperboloid of revolution about a line drawn through both sensors (a constant velocity medium is assumed). In a p-dimensional space, $N > p$ indépendent RD's are required to localize a source without additional information [1]. Accordingly, a target moving along an arbitrary path can be tracked in p dimensions with $N > p$ RD's by smoothing source location estimates obtained at various times.

Much work has been done on the problem of estimating TDOA's (a special issue on the topic is [2] and recent studies on the performance of time delay estimators include [3-10]). However, there is relatively little work on the problems of estimating source locations from TDOA measurements [11-15] and tracking moving sources using TDOA measurements from stationary passive arrays [16-19].

RD information from 1- or 2-sensor stationary passive arrays is insufficient to localize and therefore track a source moving along an arbitrary path. However, if it is assumed that the source travels along a path described by a parameter set, the problem of estimating a source's location at many points in time (i.e., the problem of tracking a source) is reduced to the problem of estimating the parameter set describing the source's path and may be possible with relatively little RD information.

Many underwater sources of interest travel along straight-line, constant-depth, constant-velocity paths, specified by so-called track parameters --

velocity, depth, bearing angle, and the radius and time of closest approach to the sensor array. In this paper, computationally efficient techniques for estimating a sources' track parameters from RD measurements taken from 1- or 2-sensor stationary passive arrays in the presence of a surface multipath reflection are presented and evaluated.

The structure of this paper is as follows. Section two describes the track parameter estimation problem and reviews basic RD relationships. The Cramer-Rao lower bound on the variance of estimating the track parameters given RD and side information is derived in Section three. Track parameter estimation methods are developed in Section four. Section five reports computer simulation results. Section six contains concluding remarks.

2. The Track Parameter Estimation Problem

We consider the following scenario illustrated in figure 1. There is a stationary passive 1- or 2-sensor array listening to a target in the presence of a surface multipath reflection. The source moves by the array at constant velocity and constant depth along a straight line. Range difference measurements (between the direct paths and/or multipaths) are available at various points during some interval of time. The track parameter estimation problem is to use the range difference measurements to estimate the parameters describing the target's path. Below, track parameters are defined and RD relationships are described for three arrays: single sensor, two-sensor vertical, and two-sensor horizontal.

Denote $\underline{L}_{s1}$ and $\underline{L}_{s2}$ as vectors containing the (x,y,z) coordinates of the sensor locations for sensors 1 and 2. Denote $\underline{p}$ as the vector of track parameters, and define

$$\underline{p} \triangleq [v_x \; v_y \; x_T \; y_T \; z_T]^T \tag{2.1}$$

Here, v_x and v_y are the x-axis and y-axis source velocities, z_T is the source depth and x_T and y_T determine the time and range of closest approach of the source to the sensor array. At any time t_i, the source location may be given as

$$\underline{L}_T = \begin{bmatrix} -t_i & 0 \\ 0 & -t_i \\ 0 & 0 \end{bmatrix} \begin{matrix} \\ \\ I_3 \end{matrix} \quad \underline{p} = [x_T - v_x t_i \;\; y_T - v_y t_i \;\; z_T] \tag{2.2}$$

where I_3 is the 3 x 3 identity matrix. The range difference between the source and the sensors at time t_i is given by

$$\underline{d}_{12}(i) = \underline{d}_{d1}(i) - \underline{d}_{d2}(i) \tag{2.3}$$

where $\underline{x}(\;)$ denotes the i-th component of the vector $\underline{x}$, the vector $\underline{d}_{12}$ contains the range differences for times t_i, i=1,...N, and $\underline{d}_{dj}(i)$ is given by

$$\underline{d}_{dj}(i) = \|\underline{L}_T(i) - \underline{L}_{sj}\| \tag{2.4}$$

Single Sensor

In the case of a single sensor in the presence of a multipath reflection, the track parameter vector, and the source and sensor locations can be given as (see figure 2)

$$\underline{\rho} = [v \; x_T \; y_T \; z_T]^T$$

$$\underline{L}_T(i) = [x_T - vt_i \;\; y_T \;\; z_T]^T$$

$$\underline{L}_s = [0 \;\; 0 \;\; z_s]^T \tag{2.5}$$

Here the x-y plane of the coordinate system is the ocean surface, positive depth is measured into the ocean, and the coordinate system origin is at the ocean surface above the sensor. Note, due to the radial symmetry of the array, there are only four elements of $\underline{\rho}$. The RD between the direct path and multipath for the sensor location above is equivalent to the RD between the direct paths to the sensor locations (see figure 2)

$$\underline{L}_{sd} = [0 \;\; 0 \;\; z_s]^T \qquad \text{(true sensor)}$$

$$\underline{L}_{sm} = [0 \;\; 0 \;\; -z_s]^T \qquad \text{(virtual sensor)}$$

The RD vector is given as

$$\underline{d} = \underline{d}_m - \underline{d}_d \tag{2.6}$$

where the i-th elements of $\underline{d}_d$ and $\underline{d}_m$, the direct and multipath ranges, are given by

$$\underline{d}_m(i) = \|\underline{L}_{sm} - \underline{L}_T(i)\|$$

$$= [(x_T - vt_i)^2 + y_T^2 + (z_T + z_s)^2]^{1/2}$$

$$\underline{d}_d(i) = \|\underline{L}_{sd} - \underline{L}_T(i)\|$$

$$= [(x_T - vt_i)^2 + y_T^2 + (z_T - z_s)^2]^{1/2} \qquad (2.7)$$

Two-sensor Vertical Array

Here, the track parameter vector and the source and sensor locations can be given as (see figure 3)

$$\underline{p} = [v \; x_T \; y_T \; z_T]^T$$

$$\underline{L}_T(i) = [x_T - vt_i \; y_T \; z_T]^T$$

$$\underline{L}_{s1} = [0 \;\; 0 \;\; z_{s1}]^T$$

$$\underline{L}_{s2} = [0 \;\; 0 \;\; z_{s2}]^T, \quad z_{s2} > z_{s1} \qquad (2.8)$$

Again, due to the radial symmetry of the array, there are only four nonzero elements of $\underline{p}$. The intersensor RD vector is given as

$$\underline{d}_{12} = \underline{d}_{d1} - \underline{d}_{d2} \qquad (2.9)$$

where the i-th elements of $\underline{d}_{d1}$ and $\underline{d}_{d2}$, the direct path ranges, are given by

$$d_{d1}(i) = \|\underline{L}_{s1} - \underline{L}_T(i)\|$$

$$= [(x_T - vt_i)^2 + y_T^2 + (z_T - z_{s1})^2]^{1/2}$$

$$d_{d2}(i) = \|\underline{L}_{s2} - \underline{L}_T(i)\|$$

$$= [(x_T - vt_i)^2 + y_T^2 + (z_T - z_{s2})^2]^{1/2} \qquad (2.10)$$

Note that this intersensor RD is the same as the multipath RD measured from a single sensor at $(0,0,\frac{1}{2}(z_{s2} - z_{s1}))$ for the same set of target track parameters with $\underline{p}(4)$ given by $z_T - \frac{1}{2}(z_{s2}+z_{s1})$ rather than z_T.

Two-sensor Horizontal Array

With a horizontal array there is no longer radial symmetry, and the track parameter vector and the source and sensor locations can be given as (see figure 4)

$$\begin{aligned}
\underline{\rho} &= [v_x \; v_y \; x_T \; y_T \; z_T]^T \\
\underline{L}_T(i) &= [x_T - v_x t_i \quad y_T - v_y t_i \quad z_T]^T \\
\underline{L}_{s1} &= [x_s \quad 0 \quad z_s]^T \\
\underline{L}_{s2} &= [-x_s \quad 0 \quad z_s]^T
\end{aligned} \tag{2.11}$$

The intersensor RD vector is given as

$$\underline{d} = \underline{d}_{d1} - \underline{d}_{d2} \tag{2.12}$$

where the ith elements of $\underline{d}_{d1}$ and $\underline{d}_{d2}$, the direct path ranges, are given by

$$\begin{aligned}
d_{d1}(i) &= \|\underline{L}_{s1} - \underline{L}_T(i)\| \\
&= [(x_T - v_x t_i - x_s)^2 + (y_T - v_y t_i)^2 + (z_T - z_s)^2]^{1/2} \\
d_{d2}(i) &= \|\underline{L}_{s2} - \underline{L}_T(i)\| \\
&= [(x_T - v_x t_i + x_s)^2 + (y_T - v_y t_i)^2 + (z_T - z_s)^2]^{1/2}
\end{aligned} \tag{2.13}$$

Note that the array is symmetric about the x-axis, also the intersensor RD is insensitive to the sign of $(z_T - z_s)$.

3. BOUNDS ON THE PERFORMANCE OF TRACK PARAMETER ESTIMATORS

In this section Cramer-Rao lower bounds on the variance of estimating track parameters from RD measurements are presented. It is shown here that the minimum variance in estimating the track parameters $\underline{p} = [v\ x_T\ y_T\ z_T]^T$ from the time history of a single multipath RD is typically large compared to the RD variance unless side information is available. It is also shown that side information may not be needed when a two sensor array is used and intersensor RD information is available.

Lower bounds on the variance in estimating the track parameters $\underline{p} = [v\ x_T\ y_T\ z_T]^T$ are presented for the case of known velocity and a single multipath RD available at N points in time. Lower bounds on the variance in estimating the track parameters $\underline{p} = [v\ x_T\ y_T\ z_T]^T$ are presented for the case of multipath and intersensor RD's available from a two-sensor vertical array at N points in time. Finally, lower bounds on the variance in estimating the track parameters $\underline{p} = [v_x\ v_y\ x_T\ y_T\ z_T]^T$ are presented for the case of the multipath and intersensor RD's available from a two-sensor horizontal array at N points in time.

The variance of an unbiased estimator of a parameter $\underline{\theta}$ is bounded below by the Cramer-Rao lower bound [20-23]:

$$\text{var}(\hat{\underline{\theta}}) = E(\hat{\underline{\theta}}\,\hat{\underline{\theta}}^T) - E(\hat{\underline{\theta}})E(\hat{\underline{\theta}})^T \geq I_{\underline{\theta}}^{-1} \tag{3.1}$$

where $I_{\underline{\theta}}$ is the Fisher information matrix for the parameter $\underline{\theta}$ given by [20]

$$I_{\underline{\theta}} \triangleq E\{[\frac{\partial}{\partial \underline{\theta}} \log f(\underline{\theta}|\underline{x})]^T [\frac{\partial}{\partial \underline{\theta}} \log f(\underline{\theta}|\underline{x})]\} \tag{3.2}$$

where $f(\underline{\theta}|\underline{x})$ is the conditional probability density function of $\underline{\theta}$ given $\underline{x}$, and the estimate $\hat{\underline{\theta}}$ is based on data $\underline{x}$. If the distribution on $\underline{\theta}$ is parameterized by $\underline{t}$ (e.g. $\underline{\theta} = g(\underline{t})$), then the Fisher information matrix for $\underline{t}$ can be written as [20]

$$I_{\underline{t}} = (\frac{\partial \underline{\theta}}{\partial \underline{t}})^T I_{\underline{\theta}} (\frac{\partial \underline{\theta}}{\partial \underline{t}}) \tag{3.3}$$

Finally, if $\underline{x}$ and $\underline{y}$ are drawn from independent distributions, then [20]

$$I_{\underline{x},\underline{y}} = I_{\underline{x}} + I_{\underline{y}} \tag{3.4}$$

3.1 Single Sensor Case

From Section 2 and (3.1), the Cramer-Rao lower bound for estimates of $\underline{p} = [v \; x_T \; y_T \; z_T]^T$ based on N single sensor RD measurements is

$$\text{var}(\hat{\underline{p}}) \geq I_{\underline{p}}^{-1} \tag{3.5}$$

where

$$I_{\underline{p}} = \left(\frac{\partial \underline{d}}{\partial \underline{p}}\right)^T I_{\underline{d}} \left(\frac{\partial \underline{d}}{\partial \underline{p}}\right) \tag{3.6}$$

where $\underline{d}$ is a vector composed of RD observations at times t_i, $i=1,\ldots,N$. The derivative, $\partial \underline{d}/\partial \underline{p}$ is easily evaluated from (1), and is given by

$$\frac{\partial \underline{d}}{\partial \underline{p}} = \begin{bmatrix} t_1(x_T - vt_1)\rho_1^- & -(x_T - vt_1)\rho_1^- & -y_T\rho_1^- & -z_T\rho_1^- + z_s\rho_1^+ \\ \vdots & \vdots & \vdots & \vdots \\ t_N(x_T - vt_N)\rho_N^- & (x_T - vt_N)\rho_N^- & -y_T\rho_N^- & -z_T\rho_N^- + z_s\rho_N^+ \end{bmatrix} \tag{3.7}$$

where

$$\rho_i^- = \frac{1}{d_m(i)} - \frac{1}{d_d(i)}, \quad \rho_i^+ = \frac{1}{d_m(i)} + \frac{1}{d_d(i)} \tag{3.8}$$

And, assuming the RD estimates are unbiased and Gaussian distributed with correlation matrix $R_{\underline{d}}$ the information in $\underline{d}$ is given by [20]

$$I_{\underline{d}} = R_{\underline{d}}^{-1} \tag{3.9}$$

3.1.1 Ambiguity Issues

As shown below, the Cramer-Rao lower bound in (3.5) is typically very large compared to the RD estimate variance; therefore low variance unbiased

track parameter estimates cannot be obtained without additional information.

For many sets of track parameters, during much of the observation time, the term $(x_T - vt_i)^2$ will dominate $\underline{d}_d(i)$ and $\underline{d}_m(i)$, and $1/\underline{d}_d(i)$ and $1/\underline{d}_m(i)$ can be approximated by,

$$\frac{1}{\underline{d}_d(i)} \sim \frac{1}{|x_T - vt_i|}\left(1 + \frac{k_d}{(x_T - vt_i)^2}\right)$$

$$\frac{1}{\underline{d}_m(i)} \sim \frac{1}{|x_T - vt_i|}\left(1 + \frac{k_m}{(x_T - vt_i)^2}\right) \qquad (3.10)$$

where k_d and k_m are constants depending on the track parameters and the sensor location. Using (3.10) in (3.8), we see that the four columns of $\frac{\partial \underline{d}}{\partial \underline{p}}$ are approximately dependent on the three vectors,

$$\begin{bmatrix} \frac{1}{|x_T - vt_1|^3} & \frac{t_1}{|x_T - vt_1|^3} & \frac{t_1^2}{|x_T - vt_1|^3} \\ \vdots & \vdots & \vdots \\ \frac{1}{|x_T - vt_N|^3} & \frac{t_N}{|x_T - vt_N|^3} & \frac{t_N^2}{|x_T - vt_N|^3} \end{bmatrix} \qquad (3.11)$$

Therefore, the matrix $I_{\underline{p}} = \left(\frac{\partial \underline{d}}{\partial \underline{p}}\right)^T I_{\underline{d}} \left(\frac{\partial \underline{d}}{\partial \underline{p}}\right)$ will be nearly singular and the resulting bounds on the minimum variance of track parameter estimators will be large compared to the variance in estimating the RD's.

This result can be seen by examining the target/sensor geometry. As illustrated in Figure 5, there are many sets of parameters which result in roughly the same measurements $\underline{d}$. Away from the sensor array, the hyperboloids of constant RD can be approximated as cones. As a result, the differences in $\underline{d}$ between the case of a target moving slowly close by a sensor and the case of a target moving more quickly, further from the sensor can be small compared to the deviation in estimating $\underline{d}$. Consequently, there

can be many sets of 'equivalent' track parameters for a given $\underline{d}$ and the variance in estimating the track parameter set from measurements of $\underline{d}$ alone is high.

3.1.2 *a priori* Estimate of v

Many times, an independent estimate of the target velocity is available, through doppler measurements, for instance [24]. With an *a priori* estimate of the target velocity (an unbiased Gaussian-distributed estimate with standard deviation σ_v^2 is assumed) the Cramer-Rao lower bound on the variance of estimating the track parameters is given by

$$\text{var}(\underline{\rho} \triangleq [\hat{v} \ \hat{x}_T \ \hat{y}_T \ \hat{z}_T]^T) \geq I_{\underline{\rho}}^{-1} \tag{3.12}$$

where

$$\begin{aligned} I_{\underline{\rho}} &= \left(\frac{\partial \underline{d}}{\partial \underline{\rho}}\right)^T I_{\underline{d}} \left(\frac{\partial \underline{d}}{\partial \underline{\rho}}\right) + \left(\frac{\partial v}{\partial \underline{\rho}}\right)^T I_v \left(\frac{\partial v}{\partial \underline{\rho}}\right) \\ &= \left(\frac{\partial \underline{d}}{\partial \underline{\rho}}\right)^T R_{\underline{d}}^{-1} \left(\frac{\partial \underline{d}}{\partial \underline{\rho}}\right) + \frac{1}{\sigma_v^2} \left(\frac{\partial v}{\partial \underline{\rho}}\right)^T \left(\frac{\partial v}{\partial \underline{\rho}}\right) \end{aligned} \tag{3.13}$$

where $\frac{\partial v}{\partial \underline{\rho}} = [1 \ \ 0 \ \ 0 \ \ 0]$ and $\frac{\partial \underline{d}}{\partial \underline{\rho}}$ is given above.

When σ_v^2 is small enough, the columns of $I_{\underline{\rho}}$ are no longer linearly dependent and estimates of the remaining track parameters will have variance comparable to the RD and velocity estimate variances (depending on the amount of RD data available and the track parameters, see section 5, simulations). As a result, the single-sensor track parameter estimation method described herein assumes an independent estimate of the velocity.

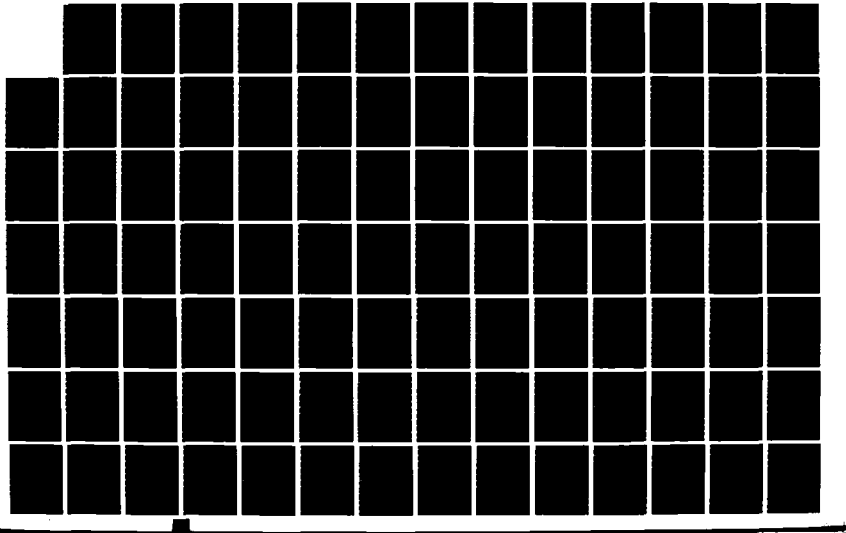
3.2 Two-sensor Vertical Array

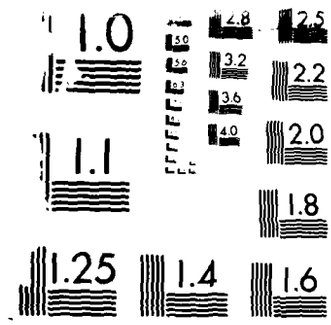
When both inter-sensor RD and multipath RD measurements are available from a two-sensor vertical array, the Cramer-Rao lower bound for estimating the set of track parameters $\underline{\rho} = [v \ x_T \ y_T \ z_T]^T$ is given by

$$\text{var}(\underline{\rho}) \geq I_{\underline{\rho}}^{-1} \tag{3.15}$$

AD-A175 078 MULTISOURCE LOCALIZATION FROM DELAY AND DOPPLER MEASUREMENTS(U) SYSTEMS CONTROL TECHNOLOGY INC PALO ALTO CA B FRIEDLANDER ET AL. JUL 86 SCT-5503 2/3

UNCLASSIFIED N00014-83-C-0727 F/G 17/7 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

where, $I_{\underline{\rho}}$ is given below using (3.3) and (3.4)

$$I_{\underline{\rho}} = I_{\underline{\rho}_1} + I_{\underline{\rho}_2} + I_{\underline{\rho}_{12}}$$
$$\triangleq \left(\frac{\partial \underline{d}_1}{\partial \underline{\rho}}\right)^T I_{\underline{d}} \left(\frac{\partial \underline{d}_1}{\partial \underline{\rho}}\right) + \left(\frac{\partial \underline{d}_2}{\partial \underline{\rho}}\right)^T I_{\underline{d}_2} \left(\frac{\partial \underline{d}_2}{\partial \underline{\rho}}\right) + \left(\frac{\partial \underline{d}_{12}}{\partial \underline{\rho}}\right) I_{\underline{d}_{12}} \left(\frac{\partial \underline{d}_{12}}{\partial \underline{\rho}}\right) \tag{3.16}$$

where $\underline{d}_1$, $\underline{d}_2$ and $\underline{d}_{12}$ are the two multipath and inter-sensor RD vectors (assumed statistically independent), the matrices $I_{\underline{d}_1}$, $I_{\underline{d}_2}$ and $I_{\underline{d}_{12}}$ are

$$I_{\underline{d}_1} = R_{\underline{d}_1}^{-1}$$

$$I_{\underline{d}_2} = R_{\underline{d}_2}^{-1}$$

$$I_{\underline{d}_{12}} = R_{\underline{d}_{12}}^{-1}$$

and the derivatives $\partial \underline{d}_1 / \partial \underline{\rho}$ and $\partial \underline{d}_2 / \partial \underline{\rho}$ are given by (3.7) and $\partial \underline{d}_{12} / \partial \underline{\rho}$ is

$$\frac{\partial \underline{d}_{12}}{\partial \underline{\rho}} = \begin{bmatrix} t_1(x_T - vt_1)\rho_1^- & -(x_T - vt_1)\rho_1^- & -y_T\rho_1^- & -z_T\rho_1^- - \frac{z_{s1}}{\underline{d}_{d1}(1)} + \frac{z_{s2}}{\underline{d}_{d2}(i)} \\ \vdots & \vdots & \vdots & \vdots \\ t_N(x_T - vt_N)\rho_N^- & -(x_T - vt_N)\rho_N^- & -y_T\rho_N^- & -z_T\rho_1^- - \frac{z_{s1}}{\underline{d}_{d1}(N)^T} + \frac{z_{s2}}{\underline{d}_{d12}(N)} \end{bmatrix} \tag{3.17}$$

where

$$\rho_i^- = \frac{1}{\underline{d}_{d1}(i)} - \frac{1}{\underline{d}_{d2}(i)}$$

Note here that, by the arguments above, the derivatives $\partial \underline{d}_1 / \partial \underline{\rho}$, $\partial \underline{d}_2 / \partial \underline{\rho}$ and $\partial \underline{d}_{12} / \partial \underline{\rho}$ each have approximately linearly dependent columns. However, unless the difference $z_{s2} - z_{s1}$ is small compared to the size of the array, the sum $I_{\underline{\rho}1} + I_{\underline{\rho}2} + I_{\underline{\rho}12}$ will have linearly independent columns, and the Cramer-Rao lower bound on the variance of the track parameters will be comparable to the RD estimate variance (depending on the track parameters and the amount of data available, see section 5, simulations).

3.2 Two-sensor Horizontal Array

When both inter-sensor RD and multipath RD measurements are available from a two-sensor horizontal array, the Cramer-Rao lower bound for estimating the set of track parameters $\underline{p} = [v_x \;\; v_y \;\; x_T \;\; y_T \;\; z_T]^T$ is given by

$$\text{var}(\underline{p}) \geq I_{\underline{p}}^{-1} \tag{3.18}$$

where, $I_{\underline{p}}$ is given as

$$I_{\underline{p}} = I_{\underline{p}_1} + I_{\underline{p}_2} + I_{\underline{p}_{12}}$$

$$\triangleq \left(\frac{\partial \underline{d}_1}{\partial \underline{p}}\right)^T I_{\underline{d}1}\left(\frac{\partial \underline{d}_1}{\partial \underline{p}}\right) + \left(\frac{\partial \underline{d}_2}{\partial \underline{p}}\right) I_{\underline{d}2}\left(\frac{\partial \underline{d}_2}{\partial \underline{p}}\right) + \left(\frac{\partial \underline{d}_{12}}{\partial \underline{p}}\right) I_{\underline{d}_{12}}\left(\frac{\partial \underline{d}_{12}}{\partial \underline{p}}\right) \tag{19}$$

where $\underline{d}_1$, $\underline{d}_2$, and $\underline{d}_{12}$ are the two multipath and inter-sensor RD functions, $I_{\underline{d}_1}$, $I_{\underline{d}_2}$ and $I_{\underline{d}_{12}}$ are

$$I_{\underline{d}_1} = R_{\underline{d}_1}^{-1}$$

$$I_{\underline{d}_2} = R_{\underline{d}_2}^{-1}$$

$$I_{\underline{d}_{12}} = R_{\underline{d}_{12}}^{-1}$$

and the derivatives $\frac{\partial \underline{d}_1}{\partial \underline{p}}$, $\frac{\partial \underline{d}_2}{\partial \underline{p}}$, and $\frac{\partial \underline{d}_{12}}{\partial \underline{p}}$ are given by

$$\frac{\partial \underline{d}_1}{\partial \underline{p}} = \begin{bmatrix} t_1(x_T - v_x t_1 - x_s)\rho_1^- & t_1(y_T - v_T t_i)\rho_1^- & -(x_T - v_x t_1 \pm x_s)\rho_1^- & -(y_T - v_y t_1)\rho_1^- & -z_T\rho_1^- + z_x\rho_1^+ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_N(x_t - v_x t_N - x_s)\rho_N^- & t_N(v_T - v_y t_i)\rho_N^- & -(x_T - v_x t_N \pm x_s)\rho_N^- & -(y_T - v_y t_N)\rho_N^- & -z_T\rho_N^- + z_s\rho_N^+ \end{bmatrix}$$

$$\frac{\partial \underline{d}_2}{\partial \underline{p}} = \begin{bmatrix} t_1(x_T - v_x t_1 + x_s)\rho_1^- & t_1(y_T - v_T t_i)\rho_1^- & -(x_T - v_x t_1 \pm x_s)\rho_1^- & -(y_T - v_y t_1)\rho_1^- & -z_T\rho_1^- + z_x\rho_1^+ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_1(x_t - v_x t_N + x_s)\rho_N^- & t_1(y_T - v_y t_i)\rho_N^- & -(x_T - v_x t_N \pm x_s)\rho_N^- & -(y_T - v_y t_N)\rho_N^- & -z_T\rho_N^- + z_s\rho_N^+ \end{bmatrix}$$

$$\frac{\partial \underline{d}_{12}}{\partial \underline{p}} = \begin{bmatrix} t_1(x_T - v_x t_1)\rho_1^- - t_1 x_s \rho_1^+ & t_1(y_T - v_y t_1)\rho_1^- & -(x_T - v_x t_1)(\rho_1^- + x_s\rho_1^+ & -(y_T - v_y t_N)\rho_1^- & -(z_T - z_s)\rho_1^- \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ t_N(x_T - v_x t_N)\rho_N^- - t_N x_s \rho_N^+ & t_N(y_T - v_y t_N)\rho_N^- & -(x_T - v_x t_N)\rho_N^- + x_s\rho_N^+ & -(y_T - y t_N)\rho_N^- & -(z_T - z_s)\rho_N^- \end{bmatrix}$$

(3.20)

Provided $\underline{x}_s$ is large enough, we expect the sum $I_{\underline{d}_1} + I_{\underline{d}_2} + I_{\underline{d}_{12}}$ will not contain linearly dependent terms and the Cramer-Rao lower bound on the track parameter estimate variance will be comparable to the RD variance. (depending on the track parameters and the amount of data available, see section 5, simulations).

4. Track Parameter Estimation

In this section closed-form equation-error methods are developed for estimating track parmeters from a single multipath RD and *a priori* knowledge of source velocity. These methods are extended to estimate track parameters using intersensor RD measurements from 2-sensor vertical and horizontal arrays. Finally, methods are presented for estimating track parameters from intersensor and multipath RD measurements from 2-sensor vertical and horizontal arrays without *a priori* knowledge the source velocity.

4.1 Single-Sensor Track Parameter Estimation: The Equation-Error Approach

In the case of a single sensor in the presence of a multipath reflection, the functional form of the multipath RD is known in terms of the track parameters. Therefore, a fit of a model $\hat{\underline{d}}$ based on track parameters $\hat{\underline{p}}$ can be made to the measured $\underline{d}$. The parameter estimates would then be chosen as

$$\hat{\underline{p}} = \min_{\underline{p}} J(\hat{\underline{d}}, \underline{d}) \tag{4.1}$$

for some cost function J.

The minimization (4.1) is over a cost function which is typically non-convex in $\underline{p}$. Therefore, in general, computationaly expensive exhaustive search methods must be used in finding a solution of (4.1). In this case the computational burden can be overcome by choosing the cost function J so that functions of the track parameters $\underline{p}$ appear as coefficeints in a linear least-squares minimization, yielding a closed-form solution for $\hat{\underline{p}}$ based on the measured RD $\underline{d}$ the sensor location $\underline{L}_s$ and an *a priori* estimate of the velocity v.

4.1.1 The Equation Error Method

Recall,

$$\underline{r} = \underline{d}_m - \underline{d} \tag{4.2}$$

where,

$$\underline{d}_m(i) = [(x_T - vt_i)^2 + y_T^2 + (z_T + z_s)^2]^{1/2} \tag{4.3a}$$

$$\underline{d}_d(i) = [(x_T - vt_i)^2 + y_T^2 + (z_T - z_s)^2]^{1/2} \tag{4.3b}$$

What is desired is an expression relating functions of $\underline{d}(i)$ to $\underline{d}_m^2(i)$ and $\underline{d}_d^2(i)$, so that functions of the unknown track parameters appear as linear coefficients which can be estimated using least-squares techniques. Manipulating (4.2) gives the following relationship

$$[\underline{d}^2(i) - \underline{d}_m^2(i) - \underline{d}_d^2(i)]^2 - 4\underline{d}_m^2(i)\underline{d}_d^2(i) = 0 \tag{4.4}$$

In (4.4) $\underline{d}_m(i)$ and $\underline{d}_d(i)$ are replaced with $\hat{\underline{d}}_m(i)$ and $\hat{\underline{d}}_d(i)$, the values predicted by the estimated track parameters $\hat{\underline{p}} = [\hat{v}\ \hat{x}_T\ \hat{y}_T\ \hat{z}_T]$, and as the delays $\underline{d}$ are not known precisely, an equation error is introduced

$$[\underline{d}^2(i) - \hat{\underline{d}}_m^2(i) - \hat{\underline{d}}_d^2(i)]^2 - 4\hat{\underline{d}}_m^2(i)\hat{\underline{d}}_d^2(i) = \varepsilon_i \tag{4.5}$$

We assume an a priori estimate of the velocity (denoted by v) is available, and assign $\hat{v} = v$. Using (4.3) in (4.5),

$$\underline{d}^4(i) - \underline{d}^2(i)(z_s^2 + t_i^2 v^2)$$

$$+ 8\underline{d}^2(i)t_i v\{\hat{x}_T\} - 4\underline{d}^2(i)\{\hat{x}_T^2 + \hat{y}_T^2 + \hat{z}_T^2\} + 16z_s^2\{\hat{z}_T^2\} = \varepsilon_i \tag{4.6}$$

where the remaining track parameters to be estimated are $[\hat{x}_T\ \hat{y}_T\ \hat{z}_T]^T$, $\underline{d}(i)$ is the measured RD at time t_i, and $\underline{\varepsilon}_i$ is the corresponding equation error [25]. Defining $\underline{\varepsilon}$ as the vector with i-th element $\underline{\varepsilon}_i$, (4.6) becomes

$$\underline{q}\ S - \underline{r} = \underline{\varepsilon} \tag{4.7a}$$

where,

$$\underline{q} = \begin{bmatrix} \hat{x}_T \\ \hat{x}_T^2 + \hat{y}_T^2 + \hat{z}_T^2 \\ \hat{z}_T^2 \end{bmatrix} \tag{4.7b}$$

$$\underline{r} \triangleq \begin{bmatrix} \underline{d}^4(1) - 4\underline{d}^2(1)(z_s^2 + t_1^2 v^2) \\ \vdots \\ \underline{d}^4(N) - 4\underline{d}^2(N)(z_s^2 + t_N^2 v^2) \end{bmatrix} \tag{4.7c}$$

$$S \triangleq \begin{bmatrix} -8\underline{d}^2(1)t_1 v & 4\underline{d}^2(1) & -16z_s^2 \\ \vdots & \vdots & \vdots \\ -8\underline{d}^2(N)t_N v & 4\underline{d}^2(N) & -16z_s^2 \end{bmatrix} \tag{4.7d}$$

Defining the cost function J as

$$J \triangleq \underline{\varepsilon}^T W \underline{\varepsilon} \tag{4.8}$$

where W is a positive-definite weighting matrix, the set of track parameters minimizing J is given by

$$\hat{\underline{p}} = \begin{bmatrix} v \\ q_1 \\ (q_2 - q_1^2 - q_3)^{1/2} \\ q_3^{1/2} \end{bmatrix} \tag{4.9}$$

where

$$\underline{q} = (S^T W S)^{-1} S^T W \underline{r}$$

Clearly, the track parameter estimates given in (4.9) are obtained with little computational expense compared to global search methods.

The estimate $\hat{p}$ given in (4.9) will be referred to as the equation-error estimate. The following sections discuss properties of the equation-error estimate and present methods for extending the equation-error estimate to the case of intersensor RD data from 2-sensor vertical and horizontal arrays.

4.1.2 The Equation Error

Here we study the cost function minimized in obtaining the equation-error

estimate. Using equations (4.4) and (4.6) the i-th equation error can be written as,

$$\underline{\varepsilon}(i) = [\underline{d}^2(i) - \hat{\underline{d}}^2(i) + 2\hat{\underline{d}}_m(i)\hat{\underline{d}}_d(i)]^2 - 4\underline{d}_m^2(i)\underline{d}_d^2(i) \tag{4.10}$$

where $\underline{d}$ is the measured RD value, $\hat{\underline{d}}$ is the RD estimated by the track parameter estimates, and $\hat{\underline{d}}_d$ and $\hat{\underline{d}}_m$ are the direct and multipath ranges given by the track parameter estimates. Rearrainging (4.10) gives

$$\underline{\varepsilon}(i) = [2\hat{\underline{d}}_m(i)\hat{\underline{d}}_d(i) + (\underline{d}^2(i) - \hat{\underline{d}}^2(i))](d^2(i) - \hat{d}^2(i)) \tag{4.11}$$

The factor $\hat{\underline{d}}^2(i) - \underline{d}^2(i)$ is the difference between the square measured and estimated RD's, and should be much smaller than $\hat{\underline{d}}_m(i)\hat{\underline{d}}_d(i)$, the product of the estimated direct and multipath ranges. Therefore, the i-th equation error can be apporximated as

$$\begin{aligned}\underline{\varepsilon}(i) &\approx 2\hat{\underline{d}}_m(i)\hat{\underline{d}}_d(i)(\hat{d}^2(i) - \hat{d}^2(i)) \\ &= \underline{w}(i)(\underline{d}(i) - \hat{d}(i))\end{aligned} \tag{4.12}$$

where

$$\underline{w}(i) \triangleq 2\hat{\underline{d}}_m(i)\hat{\underline{d}}(i)(\hat{\underline{d}}(i) + \underline{d}(i))$$

Note that the cost function $J = \underline{\varepsilon}^T\underline{\varepsilon}$ is made small by selecting track parameters which have $\|\underline{w}\|$ as small as possible such that $\hat{\underline{d}}$ approximates $\underline{d}$. The factor $\underline{w}(i)$ is made smaller by selecting track parameters which place the source closer to the point (0,0,0), recall $\underline{L}_s = [0\ 0\ z_s]^T$. Thus, it is expected that minimizing the unweighted equation-error (minimizing $J = \underline{\varepsilon}^T\underline{\varepsilon}$) will produce track parameter estimates with a bias in the direction $[\hat{x}_T\ \hat{y}_T\ \hat{z}_T] = [0\ \ 0\ \ 0]$.

Minimizing $J = \underline{\varepsilon}^T\underline{\varepsilon}$ is equivalent to minimizing a weighted L_2 norm of the difference of the measured RD values and the estimated RD values. Specificly, the minimization $J = \underline{\varepsilon}^T\underline{\varepsilon}$ is equivalent to the minimization

$$J = \underline{\varepsilon}^{*T}\,\Lambda\,\underline{\varepsilon}^*$$

where $\underline{\varepsilon}^* \triangleq \underline{d} - \hat{\underline{d}}$ and $\mathbf{A}$ is a diagonal matrix with i-th row/column entry given as $\underline{w}^2(i)$.

If it is desired to minimize $J^* = \underline{\varepsilon}^{*T}\underline{\varepsilon}$, the above suggests a weighting matrix and an iterative procedure for doing so: solve (4.7) iteratively with weighting matrix

$$W_{n+1} = \text{diag}\{1/\underline{w}^2(i)\} \tag{4.13}$$

where diag$\{\underline{x}(i)\}$ denotes a diagonal matrix with $\underline{x}(i)$ as the i-th row/column entry, and $\underline{w}(i)$ is computed using track parameter estimates from the n-th solution of (4.7) minimizing $J = \underline{\varepsilon}^T W_n \underline{\varepsilon}$. If the process converges, the error minimized is

$$J^* = \underline{\varepsilon}^{*T}\underline{\varepsilon}^* \tag{4.14}$$

Note that minimizing J^* rather than J places more relative weight on the RD estimates taken when the source is closer to the sensor. The benefit of minimizing J^* over J may be evaluated by the studying the Fisher information matrix (3.12) for the relative importance of RD measurements as a function of source range.

4.1.2 Variance of the Equation-Error Estimate

In this section we present expressions for the variance of the unweighted equation-error track parameter estimates when RD and a priori velocity information is noisy.

Velocity Known Precisely

We first derive an expression for the equation-error estimate variance when the velocity is known precisely and the RD vector is unbiased with variance $R_{\underline{d}}$. When the RD standard deviation is small, the variance of the track parameters can be given by [26]

$$\text{var}(\hat{\underline{p}} = [\hat{v}\ \hat{x}_T\ \hat{y}_T\ \hat{z}_T]^T) \approx \left(\frac{\partial \underline{p}}{\partial \underline{d}}\right) R_{\underline{d}} \left(\frac{\partial \underline{p}}{\partial \underline{d}}\right)^T \tag{4.15}$$

where $\partial \underline{p}/\partial \underline{d}$ can be evaluated by the chain rule:

$$\frac{\partial \underline{p}}{\partial \underline{d}} = \left(\frac{\partial \underline{p}}{\partial \underline{q}}\right)\left(\frac{\partial \underline{q}}{\partial \underline{d}}\right) \tag{4.16}$$

From (4.7), $\frac{\partial \underline{p}}{\partial \underline{q}}$ is given as

$$\frac{\partial \underline{p}}{\partial \underline{q}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ -\frac{p_1}{p_2} & \frac{1}{2p_2} & -\frac{1}{2p_2} \\ 0 & 0 & \frac{1}{2p_3} \end{bmatrix} \tag{4.17}$$

where p_i is the i-th element of $\underline{p}$. Recall in the noiseless case,

$$S\,\underline{q} = \underline{r} \tag{4.18}$$

Differentiating (4.18) by $\underline{d}$ gives

$$\frac{\partial S}{\partial \underline{d}}\,\underline{q} + S\,\frac{\partial \underline{q}}{\partial \underline{d}} = \frac{\partial \underline{r}}{\partial \underline{q}} \tag{4.19}$$

and solving for $\frac{\partial \underline{q}}{\partial \underline{d}}$,

$$\frac{\partial \underline{q}}{\partial \underline{d}} = (S^T S)^{-1} S^T \left\{ -\frac{\partial S}{\partial \underline{d}}\,q + \frac{\partial \underline{r}}{\partial \underline{d}} \right\} \tag{4.20}$$

where

$$\frac{\partial \underline{r}}{\partial \underline{d}} = \mathrm{diag}\{4\underline{d}^3(i) + 8\underline{d}(i)(v^2 t_i^2 + z_s^2)\}$$

$$\frac{\partial S}{\partial \underline{d}}\,\underline{q} = \mathrm{diag}\{-16\underline{d}(i)t_i v q_1 + 8\underline{d}(i)q_2\} \tag{4.21}$$

RD's Known Precisely

When the RD's are known precisely and the velocity estimate v is unbiased with small variance σ_v^2, the equation-error estimate variance can be derived in a manner similar to (4.15)-(4.21):

$$\text{var}(\underline{\rho} = [\hat{v}\ \hat{x}_T\ \hat{y}_T\ \hat{z}_T]) \sim (\frac{\partial \underline{\rho}}{\partial v})\sigma_v^2(\frac{\partial \underline{\rho}}{\partial v})^T \tag{4.22}$$

where $\frac{\partial \underline{\rho}}{\partial v}$ can be evaluated by the chain rule:

$$\frac{\partial \underline{\rho}}{\partial v} = (\frac{\partial \underline{\rho}}{\partial \underline{q}})(\frac{\partial \underline{q}}{\partial v}) + \sigma_v^2[1\ \ 0\ \ 0\ \ 0]^T \tag{4.23}$$

the derivative $\frac{\partial \underline{\rho}}{\partial \underline{q}}$ is given in (4.17) and the derivative $\frac{\partial \underline{q}}{\partial v}$ can be evaluated as follows. In the noiseless case,

$$S\,\underline{q} = \underline{r} \tag{4.24}$$

Differentiating (4.24) by v gives

$$\frac{\partial S}{\partial v}\,\underline{q} + S\,\frac{\partial \underline{q}}{\partial v} = \frac{\partial \underline{r}}{\partial v} \tag{4.25}$$

and solving for $\frac{\partial \underline{q}}{\partial v}$,

$$\frac{\partial \underline{q}}{\partial v} = (S^T S)^{-1} S^T \{-\frac{\partial S}{\partial v}\,\underline{q} + \frac{\partial \underline{r}}{\partial v}\} \tag{4.26}$$

where

$$\frac{\partial \underline{r}}{\partial v} = 8v \begin{bmatrix} \underline{d}^2(1)t_1 \\ \vdots \\ \underline{d}^2(N)t_N \end{bmatrix}$$

$$\frac{\partial S}{\partial v}\,\underline{q} = \begin{bmatrix} \underline{d}^2(1)t_1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ \underline{d}^2(N)t_N & 0 & 0 \end{bmatrix} 8\underline{q} \tag{4.27}$$

Estimated RD's and Velocity

Finally, if the estimates of $\underline{d}$ and v are independent, then the variance of the equation error estimate $\hat{\rho} = [\hat{v}\ \hat{x}_T\ \hat{y}_T\ \hat{z}_T]$ can be given approximately as the sum of the variances calculated above

$$\text{Var}(\underline{\rho}) \sim \sigma_v^2(\frac{\partial \underline{\rho}}{\partial v})(\frac{\partial \underline{\rho}}{\partial v})^T + (\frac{\partial \underline{\rho}}{\partial \underline{d}})R_d(\frac{\partial \underline{\rho}}{\partial \underline{d}})^T \tag{4.28}$$

4.2 Two Sensor Track Parameter Estimation

In this section methods for adapting the equation-error estimate to inter-sensor RD measurements from two sensor arrays are presented. In addition, methods for estimating track parameters from two-sensor arrays without *a priori* information are developed.

4.2.1 Track Parameter Estimation From Inter-sensor RD Estimates

The modification of the equation-error estimate (4.9) to make use of inter-sensor delay information rather than multipath delay information is straight forward. If the sensors are placed in a vertical array, the intersensor RD is functionally equivalent to the single sensor multipath RD, and only the target and sensor depths need to be adjusted:

$$\hat{\underline{p}} = \begin{bmatrix} \hat{v} \\ \hat{x}_T \\ \hat{y}_T \\ \hat{z}_T \end{bmatrix} \quad \begin{bmatrix} v \\ q_1 \\ (q_2 - q_3 - q_1^{1/2} \\ q_3^{1/2} + \frac{1}{2}(z_{S2} - z_{S1}) \end{bmatrix} \tag{4.29}$$

where

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = (S^T W S)^{-1} S^T W \underline{r}$$

where S and $\underline{r}$ are as defined in (4.7) with $\underline{d} = \underline{d}_{12}$ the intersensor RD estimate and W is a positive definite weighting matrix.

In the case of a horizontal array, an equation-error estimate can be derived in a manner similiar to (4.2) -- (4.9). From (2.12),

$$\{\underline{d}_{12}^2(i) - \underline{d}_{d1}^2(i) - \underline{d}_{d2}^2(i)\}^2 = 4\underline{d}_{d1}^2(i)\,\underline{d}_{d2}^2(i) \tag{4.30}$$

where $\underline{d}_{12}$ is the intersensor RD, $\underline{d}_{d1}$ is the target range to sensor 1 and $\underline{d}_{d2}$ is the target range to sensor 2.

Substituting (2.13) into equation (4.30),

$$\underline{d}_{12}^4(i) - 4\underline{d}_{12}^2(i)\{(x_T - v_x t_i - x_s)^2 + 2x_s(x_T - v_x t_i) + (y_T - v_y t_i) + (z_s - z_T)^2\} + 4x_s(x_T - v_x t_i) = 0 \tag{4.31}$$

Denoting the track parameters to be estimated by $\hat{p} = [\hat{v}_x \ \hat{v}_y \ \hat{x}_T \ \hat{y}_T \ \hat{z}_T]^T$ and $\underline{d}_{12}$ as the measured inter-sensor RD, (4.31) becomes

$$\underline{d}_{12}^4(i) - 4\underline{d}_{12}^2(i)[t_i^2 v^2 + x_s^2] + 8\underline{d}_{12}^2(i)t_i\{\hat{v}_x\hat{x}_T + \hat{v}_y\hat{y}_T\} - 4\underline{d}_{12}^2(i)\{\hat{x}_T^2 + \hat{y}_T^2 + (\hat{z}_T - z_s)^2\} - 4x_s t_i\{\hat{v}_x\} + 4x_s\{\hat{x}_T\} = \underline{\varepsilon}_i \tag{4.32}$$

where an equation error $\underline{\varepsilon}_i$ has been introduced. The set of track parameters minimizing $J = \underline{\varepsilon}^T W \underline{\varepsilon}$ are given by

$$\hat{p} = \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ \hat{x}_T \\ \hat{y}_T \\ (z_s - \hat{z}_T)^2 \end{bmatrix} = \begin{bmatrix} q_3 \\ (v^2 - q_3^2)^{1/2} \\ q_4 \\ (q_1 - q_3 q_2)/(v^2 - q_3^2)^{1/2} \\ q_2 - q_4^2 - (q_1 - q_3 q_2)/(v^2 - q_3^2)^{1/2} \end{bmatrix} \tag{4.33}$$

where $v = (v_x^2 + v_y^2)^{1/2}$ is the a priori velocity estimate and

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = (S^T W S)^{-1} S^T W \underline{r}$$

where

$$S \triangleq \begin{bmatrix} -8\underline{d}_{12}^2(1)t_1 & 4\underline{d}_{12}^2(1) & 4x_s t_1 & 4x_s \\ \vdots & \vdots & \vdots & \vdots \\ -8\underline{d}_{12}^2(N)t_N & 4\underline{d}_{12}^2(N) & 4x_s t_1 & 4x_s \end{bmatrix}$$

$$\underline{r} \triangleq \begin{bmatrix} \underline{d}_{12}^4(1) - 4\underline{d}_{12}^2(1)(t_1^2 v^2 + x_s^2) \\ \vdots \\ \underline{d}_{12}^4(N) - 4\underline{d}_{12}^2(N)(t_N^2 v^2 + x_s^2) \end{bmatrix}$$

and W is a positive definite weighting matrix. Note that due to symmetries in the array, the signs of y_T and v_y, and $(z_T - z_s)$ cannot be determined from RD information alone; however, the sign of $y_T v_T$ can be estimated, and here, $\hat{v}_y > 0$ and $\hat{y}_T$ takes the sign of $\hat{y}_T \hat{v}_y$

4.2 Combining estimates from two sensors

As shown in section 3, when RD information from more than one sensor is available, side information may not be needed to obtian low variance track parameter estimates. Below methods for estimating track parameters from inter-sensor and multipath RD information from two-sensor vertical and horizontal arrays are given. These methods are not optimal and do not produce ML (minimum variance, unbiased) track parameter estimates, but provide a computationally inexpensive alternative to the nonlinear optimization involved in computing the ML track parameter estimates. If the resources are available to compute the ML estimate via an iterative nonlinear minimization, for example, these methods provide excellent starting points.

With data available from two sensors placed in a vertical array, sets of

estimates, parameterized by v can be constructed using (4.9) and (4.29). A line search over v can then be performed for the set of track parameters minimizing some cost function. Emperically, it was found that minimizing the following cost function gave relatively unbiased low-variance estimates for a large range of noise levels for the track parameter sets used in the Monte-Carlo simulations, see Section 5:

$$J_v = (\hat{z}_{T12}(v) - \hat{z}_{T2}(v))^2 \tag{4.34}$$

where $\hat{z}_{T12}(v)$ is the depth estimate based on v using the intersensor RD information, and $\hat{z}_{T2}(v)$ is the depth estiamte based on v using multipath RD information from the deeper sensor.

If the data is from two sensors placed in a horizontal array, sets of estimates, parameterized by v based on individual sensor data can be generated by (4.9). Estimates of $\hat{v}_x/v$ parameterized by v can be made based on inter-sensor data form (4.33). Track parameter estimates can then be chosen by finding the set of estimates producing the most consistent $\hat{y}_{T1}$, $\hat{y}_{T2}$, $\hat{v}_x/v$, i.e. the set, minimizing the cost function

$$J_H = \left(\frac{\hat{v}_x(v)}{v} - \frac{\hat{y}_{T1}(v)-\hat{y}_{T2}(v)}{2x_s}\right)^2 \tag{4.35}$$

where $\hat{v}_x(v)$ is the x-axis velocity estimate obtained from the intersensor RD estimate using (4.33) and $\hat{y}_{T1}$ and $\hat{y}_{T2}$ are the y-axis range estimates from the multipath RD measurements and (4.9).

5. Simulation Results

This section reports computer simulation results on the performance of equation-error estimates applied to noisy RD and velocity estimates. Sample bias, standard deviation and RMS error, defined by

$$\text{sample bias} \triangleq \frac{1}{N}\sum_{i=1}^{N}(\hat{\theta}_i - \theta)$$

$$\text{sample standard deviation} \triangleq \left[\frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N}\hat{\theta}_i\right)^2\right]^{1/2}$$

$$\text{sample RMS error} \triangleq \left[\frac{1}{N}\sum_{i=1}^{N}(\hat{\theta}_i - \theta)^2\right]^{1/2} \qquad (5.1)$$

(where $\hat{\theta}_i$ is the i-th sample estimate of the parameter θ)were calculated by averaging results of 100-trial Monte-Carlo runs and were compared to theoretically calculated values and Cramer-Rao lower bounds. Simulations were implemented in the Ctrl-C* language on a VAX 11/785 computer.

Results are given for sources moving along straight-line constant-velocity paths by 1- and 2-sensor arrays. Multipath and inter-sensor RD estimates were calculated by adding white Gaussian noise to the true RD vectors; velocity estimates (used in the 1-sensor simulations) were calculated by adding white Gaussian noise to the true velocity.

Table 1 describes the enviromental data (sensor locations, track parameters noise levels, etc.) for each of the Monte-Carlo runs (the source trajectories and sensor locations are also shown in Figure 6). Note that in the case of a single sensor or vertical array positioned at the origin, $\underline{\rho} = [v_x \; v_y \; x_T \; y_T \; z_T]^T = [5 \;\; 2 \;\; 250 \;\; 900 \;\; 170]^T$ is equivalent to $\underline{\rho} = [v \; x_T \; y_T \; z_T]^T = [5.38 \;\; 556.3 \;\; 747.8 \;\; 170]^T$; and $[-3 \;\; 1 \;\; 140 \;\; 400 \;\; 110]^T$ translates to $[3.162 \;\; 8.849 \;\; 423.7 \;\; 110]^T$. Each run used 100 points of data: one RD sample every 10 seconds from time $t_1 = -490$ to $t_{100} = 500$ seconds, and the total observation time is $t = t_{100} - t_1 = 990$ seconds. So

*Ctrl-C is a high-level matrix calculator language sold by SCT, Inc.

that all biases and deviations are measured in meters, the track parameter velocity is replaced here by the distance travelled during the observation interval vt.

Single sensor

Table 2 shows sample bias, standard deviation, RMS error, and theoretical deviation (4.28) of the unweighted equation-error estimate (4.9) applied to multipath RD estimates and velocity estimates from a single sensor (Monte-Carlo runs 1 thru 4). Monte-Carlo runs 1 and 2 show the case of perfectly known RD estimates and noisy velocity estimates, and runs 3 and 4 show the case of noisy RD estimates and perfectly known velocity.

In these cases, the equation-error estimator is seen to be essentially unbiased and to exhibit a standard deviation comparable to the RD and velocity deviations. Note that the theoretical deviations given by equation (4.28) agree very well with the sample RMS errors. Accordingly, it was noted that the track parameter estimate standard deviation appears to increase linearly with RD and velocity standard deviation. As expected (by the discussion of the equation error, Section 4.1), the sample bias of the equation-error estimate is negative in runs 3 and 4, indicating that the track parameter estimate pulls the estimated source location closer to the origin than the true source location.

Table 3 shows sample RMS errors theoretical deviations (4.28), and Cramer-Rao bounds (3.12) of the equation-error estimator appled to RD and velocity estimates from a single sensor (Monte-Carlo runs 5 thru 12). The sample bias, not shown here, was noted to be small compared to the sample standard deviation. The RMS error appears to be slightly smaller than theroetically predicted, consistent with a possible small cross-correlation between the RD and velocity noises seen in these relatively small sample-size Monte-Carlo runs. The sample RMS errors (and theoretical deviations) appear to be very close to the Cramer-Rao bounds indicating that the equation-error estimate is using the RD and velocity information in an efficient manner for these cases.

Two-snesor Vertical Array

Table 4 shows sample bias, sample standard deviation, sample RMS error and Cramer-Rao bounds (3.15) for equation-error estimates applied to RD measurements from a two-sensor vertical array; a priori velocity estimates were not given. The track parameter estimates were made from RD measurements taken from the deepest sensor with velocity estimates determined by (4.34). The equation-error estimate sample bias is seen to be a strong (perhaps quadratic) function of the the RD noise deviation, and is comparable to the sample standard deviation at the larger value of additive RD noise. The sample standard deviation appears to be a linear function of the RD noise deviation, and small compared to the track parameter values. The equation-error estimate sample RMS error is seen to be about twice the Cramer-Rao bound.

Two-sensor Horizontal Array

Table 5 shows sample bias, sample standard deviation, sample RMS error, and Cramer-Rao bounds for the case of the source moving past a horizontal array. The track parameters were estimated by averaging the translated track parameter estimates given by (4.9) and (4.3) with a velocity estimated by minimizing the cost function given in (4.35). Here, the track parameter estimates are essentially unbiased and have a sample standard deviation which is small compared to the track parameter value and appears to increase linearly with RD standard deviation. The estimates have an RMS error of about five to ten times the Cramer-Rao bound, depending on the track parameters. Here, it appears that the track parameter estimation method is not using the RD information efficiently. However, depending on the application this estimator might have acceptable performance.

6. Summary

In this paper, we discussed the problem of tracking sources with multipath and intersensor range difference measurements form 1- or 2-sensor stationary passive arrays. RD data gathered from such arrays are insufficient to track a source moving along an arbritrary path, and here, the problem of describing the source's location as a function of time was reduced to the problem of estimating a small set of parameters describing an assumed straight-line, constant-velocity, constant-depth source path.

Cramer-Rao bounds were presented for estimating the track parameters from the time history of multipath and intersensor range difference measurements. It is shown that this track parameter set could not be accurately estimated from the time history of a single multipath range difference without side information. However, multipath and intersensor range difference measurements from a two-sensor array were seen to be sufficient to estimate the track parameter set when the sensors are appropiately placed and enough RD data is available.

Linear least-squares equation-error techniques were presented which estimate track parameters from independent velocity estimates and multipath range difference measurements taken from a 1-sensor array. Analytic expressions were developed for the variance of the estimate, and Monte-Carlo simulations were presented which show that these estimators are relatively unbiased and have sample RMS error which is given accurately by the analytic expressions and is approximately equal to the Cramer-Rao bound.

Line-search methods were developed to estimate track parameters from 2-sensor RD data when no independent velocity estimates were available. The sample mean square error of the track parameter estimates produced by these methods was shown by Monte-Carlo simulations to be in the range of two to ten times the corresponding Cramer-Rao bounds. These methods provide a computationally inexpesive alternative to the nonlinear optimization required in finding the ML estimate.

References

[1] J. P. Van Etten, "Navigation Systems: Fundamentals of Low and Very Low Frequency Hyperbolic Techniques," Electrical Commun., vol. 45, no. 3, pp. 192--212, 1970.

[2] Special issue on time-delay estimation, IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. ASSP-29, June 1981.

[3] J.P. Ianniello, "The Variance of Multipath Time Delay Estimation Using Autocorrelation," NUSC Tech. Memo 831008, Jan. 24, 1983.

[4] _____, "Large and Small Error Performance Limits for Multipath Time Delay Estimation," IEEE Trans. on Acoust., Speech, and Signal Processing, vol. ASSP-34, Number 2, April 1986.

[5] _____, "Time Delay Estimation Via Cross-Correlation in the Presence of Large Estimation Errors," IEEE Trns. Acoust., Speech, Signal Processing, vol. ASSP-30, pp. 998-1003, 1982.

[6] A. Weiss and E. Weinstein, "Fundamental Limitations in Passive Time Delay Estimation -- Part 1: Narrow-band Systems," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 472-485, 1983.

[7] J.P. Ianniello, E. Weinstein, and A. Weiss, "Comparison of the Ziv-Zakai Lower Bound on Time Delay Estimation with Correlator Performance," in ICASSP 83 Conf. Proc., Boston, MA, Apr. 1983, pp. 875-878.

[8] _____, "Lower Bounds on Worst-Case Probability of Large Error for Two Channel Time Delay Estimation," submitted to Trans. Acoust., Speech, Signal Processing, 1984.

[9] B. Friedlander, "On the Cramer-Rao Bound for Time Delay and Doppler Estimation," IEEE Trans. Info. Theory, vol. IT-30, no. 3, pp. 575--580, May 1984.

[10] J. C. Hassab and R. Boucher, "A Probabilistic Analysis of Time Delay Extraction by the Cepstrum in Stationary Gaussian Noise," IEEE Trans. Inform. Theory, vol. IT-22, pp. 444-454, July 1976.

[11] R. O. Schmidt, "A New Approach to Geometry of Range Difference Location," IEEE Trans. Aero. and Elec. Systems, vol. AES-8, no. 6, pp. 821--835, Nov. 1972.

[12] J. M. Delosme, M. Morf, and B. Friedlander, "A Linear Equation Approach to Locating Sources from Time-Difference-of-Arrival Measurements," Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, 1980.

[13] Schau and A. Z. Robinson, "Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences," submitted for publication.

[14] J. O. Smith and J. S. Abel, "Closed-Form Least-Squares Localization of Multiple Broad-Band Emitters from Time-Difference-Of-Arrival

Measurements," submitted for publication, IEEE Trans. on Acoust., Speech, and Sig. Proc.

[15] J.S. Abel and J.O. Smith, "On the Efficiency of the Spherical Interpolation Estimator of Source Location from Range Difference Measurements," submitted, IEEE Trans. on Acost., Speech, and Sig. Proc.

[16] R.L. Moose and T.E. Dailey, "Adaptive Underwater Target Tracking Using Passive Multipath Time-Delay Measurements," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-33, No. 4, August 1985.

[17] D. H. McCabe and R.L. Moose, "Passive Source Tracking Using Sonar Time Delay Data," IEEE Trans. Acoust., Speech, Signal Processing, June 1981.

[18] R.L. Mosse, "Passive Source Tacking Using Differential Time Delays," in IEEE Proc. 1981 EASCON Conf., Nov. 1981, Washington, D.C.

[19] J.C. Hassab, "Passive Tracking of a Moving Source by a Single Observer in Shallow Water," J. Sound and Vibration, vol. 44, no. 1, 1976.

[20] E.L. Lehmann, Theory of Point Estimation, John Wiley and Sons, New York, NY, pp. 123-129, 1983.

[21] R.A. Fisher, "On the Mathematical Foundations of Theoretical Statistics," Phil. Trans. Roy. Soc., London, 222, 306, 1922.

[22] H. Cramer, Mathematical Methods of Statistics, Princeton University Press, Princeton, NJ, 1935.

[23] C.R. Rao, "Information and Accuracy Attainable in the Estimation of Statistical Parameters," Bull. Calcutta Math. Soc., vol. 37, pp. 81-91, 1945.

[24] Khosrow Lashkari, B. Friedlander and J. S. Abel, "Track Parameter Estimation Using Multipath Delay and Doppler Informations," SCT Final Report 5517, Contract No. NUUU14-84-C-0408, February, 1986.

[25] L. Ljung and T. Soderstrom, Theory and Practice of Recursive Identification, MIT Press, Cambridge MA, 1984.

[26] D. R. Cox and D. V. Hinkley, Theoretical Statistics, Chapman and Hall, London, 1982.

Figure captions

Figure 1 RD Measurement Scenario. A radiating source is shown moving along a constant-depth, constant-velocity, straight-line path. The sensor is shown recieving signals from the source both directly and from a surface multipath reflection.

Figure 2 Single Sensor RD Measurement and Source Track. Figure 2a shows the sensor at $\underline{L}_{sd}$ located on the z-axis and the source at $\underline{L}_T$. The direct path and multipath lengths are shown as d_d and d_m. Note that the multipath length d_m is the same as the direct path length from the source to a 'virtual' sensor located at $\underline{L}_{sm}$. Figure 2b shows the source's constant-velocity motion parallel the x-axis and the sensor located at the origin of the x-y plane.

Figure 3 Vertical Array RD Measurement and Source Track. Figure 3a shows sensors at $\underline{L}_{s1}$ and $\underline{L}_{s2}$ on the z-axis and the source at $\underline{L}_T$. The direct path lengths from the source to the sensors are shown as d_{d1} and d_{d2}; the multipaths are not shown. Figure 3b shows the source's constant-velocity motion parallel the x-axis and the sensors located at the origin of the x-y plane.

Figure 4 Horizontal Array RD Measurement and Source Track. Figure 4a shows sensors at $\underline{L}_{s1}$ and $\underline{L}_{s2}$ at the same depth, and the source at $\underline{L}_T$. The direct paths from the source to the sensors are shown and have lengths d_{d1} and d_{d2}; the multipaths are not shown. Figure 4b shows the source's constant-velocity motion and the sensors located on the x-axis.

Figure 5 Track Equivalence. Two sources moving along different constant-velocity, constant-depth, straight-line paths are shown here to produce similiar sets of single-sensor multipath RD measurements. Source 1 is shown moving along a constant-velocity, constant-depth, straignt-line path producing RD measurements $[d_1\ d_2\ d_3]$ at times $[t_1\ t_2\ t_3]$. Source 2 is shown moving faster at greater depth along a constant-velocity, constant-depth,

straight-line path producing the same RD measurements at the same times.

Figure 6 Simulation Source Tracks. The x-y plane source tracks used in the Monte-Carlo simulations are shown as dotted lines, each dot representing the source location at the time of an RD measurement. The sensor locations in the x-y plane are shown as squares; the vertical array and single-sensor array are represented by the center square and the horizontal array is represented by the two outer squares.

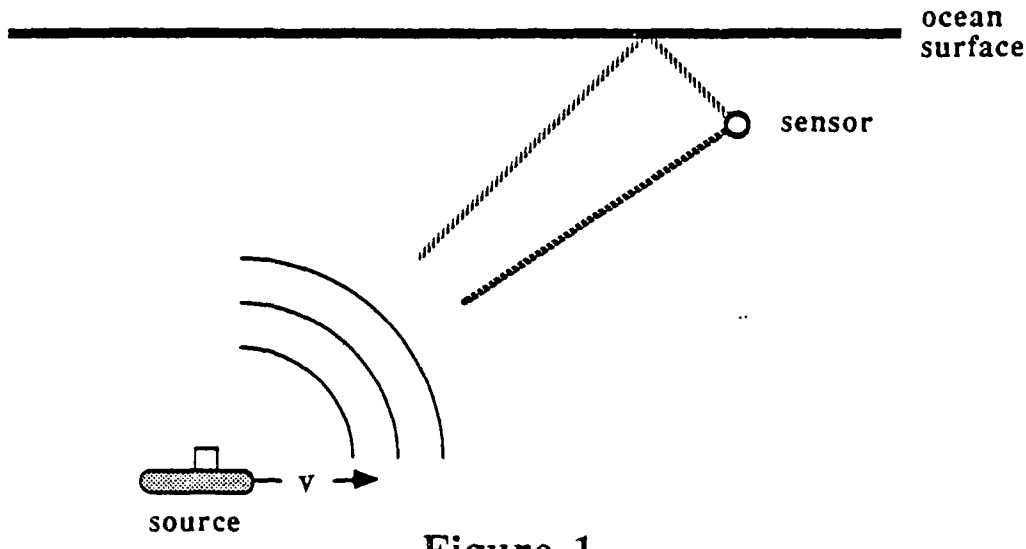


Figure 1

Side view

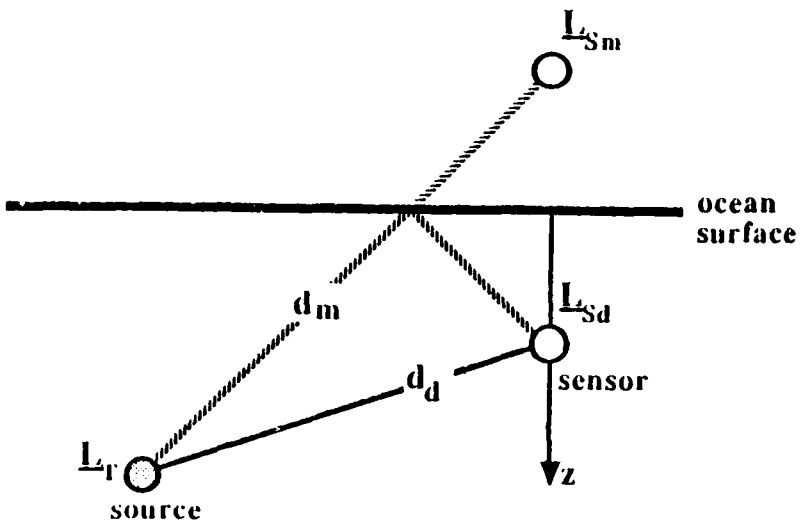


Figure 2a

Top view

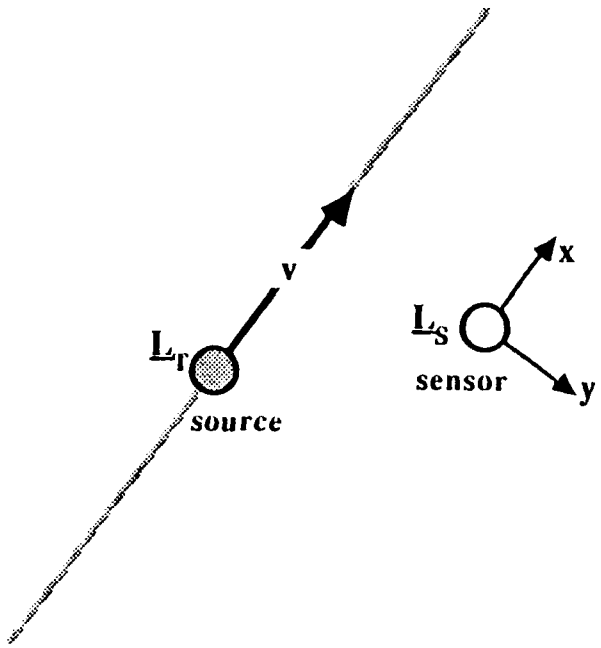


Figure 2b

Side view

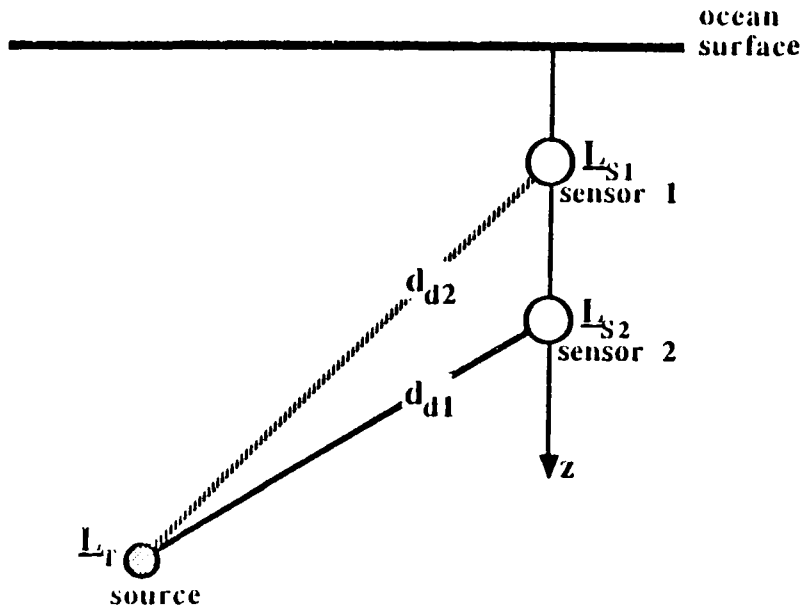


Figure 3a

Top view

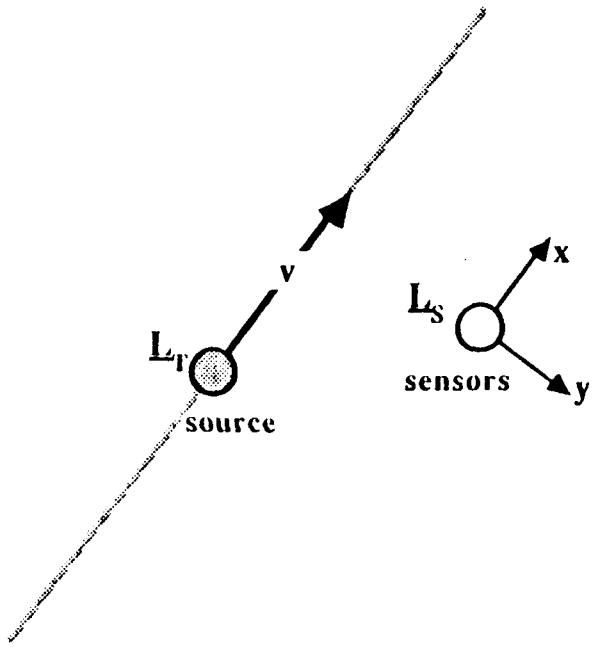


Figure 3b

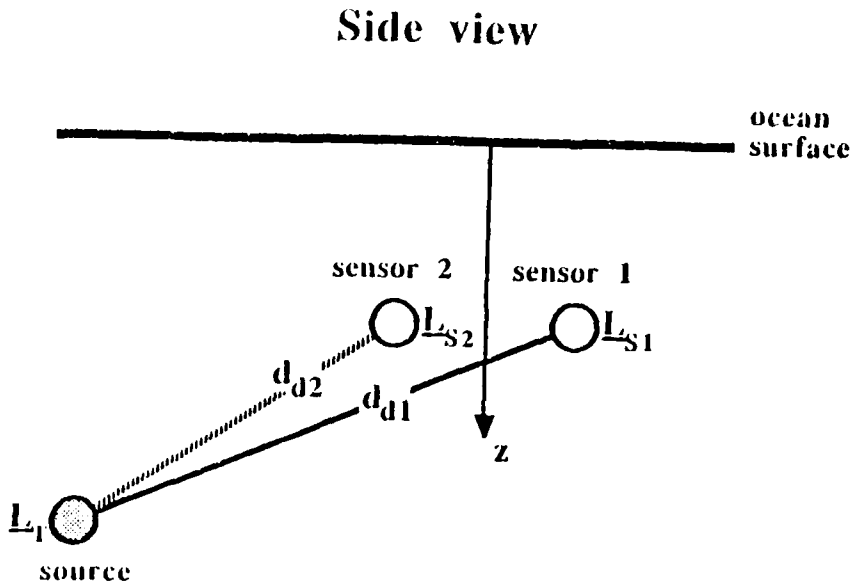


Figure 4a

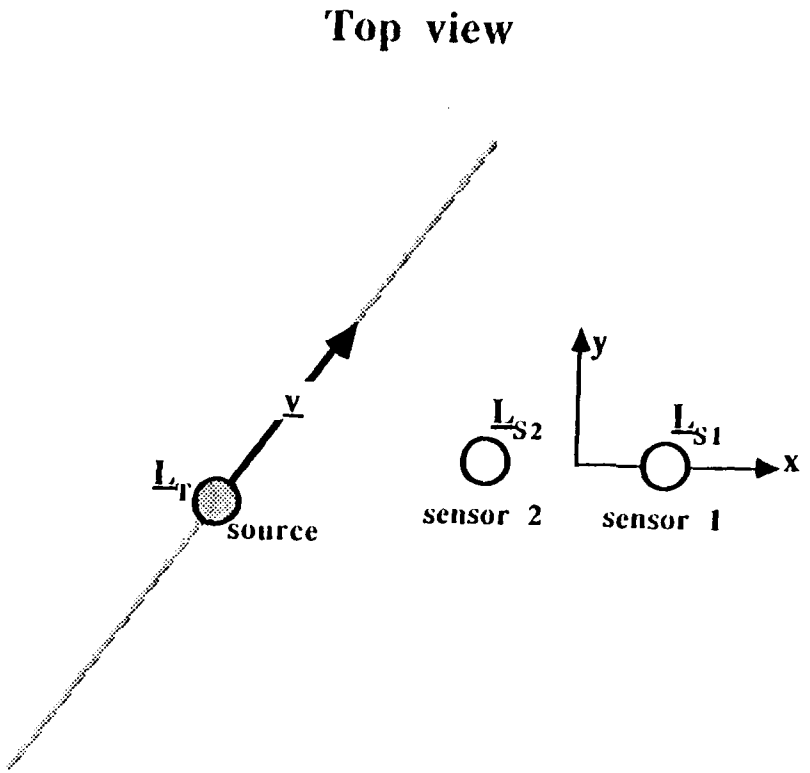


Figure 4b

Side view

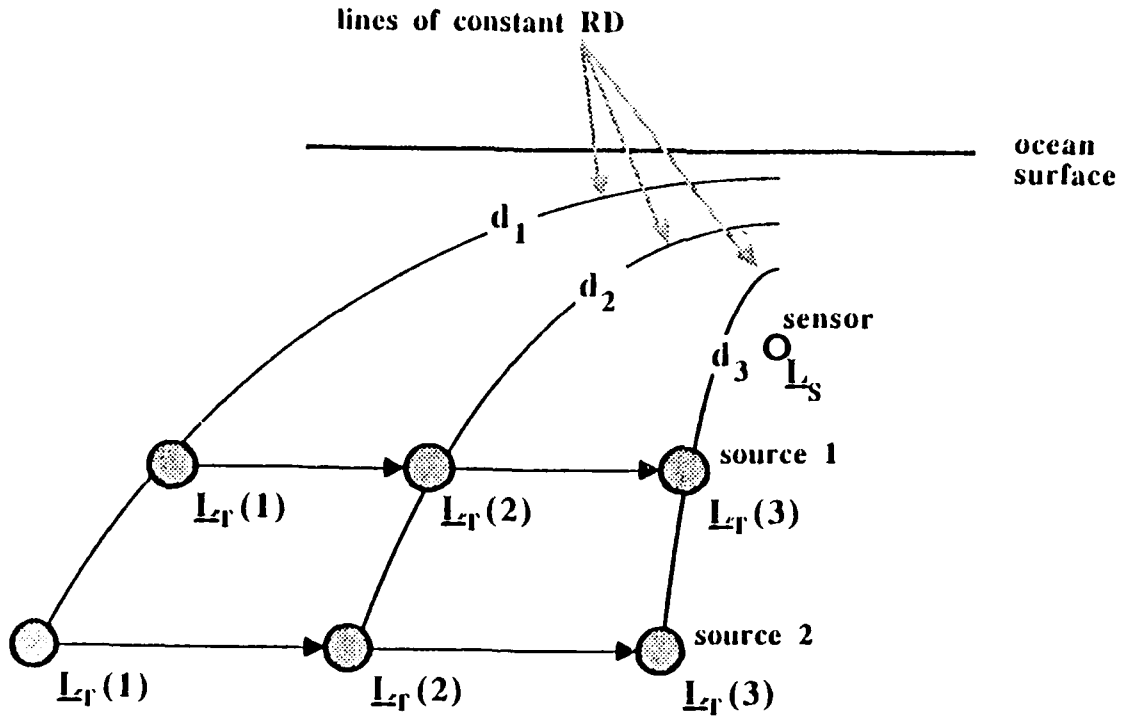


Figure 5

Figure 6

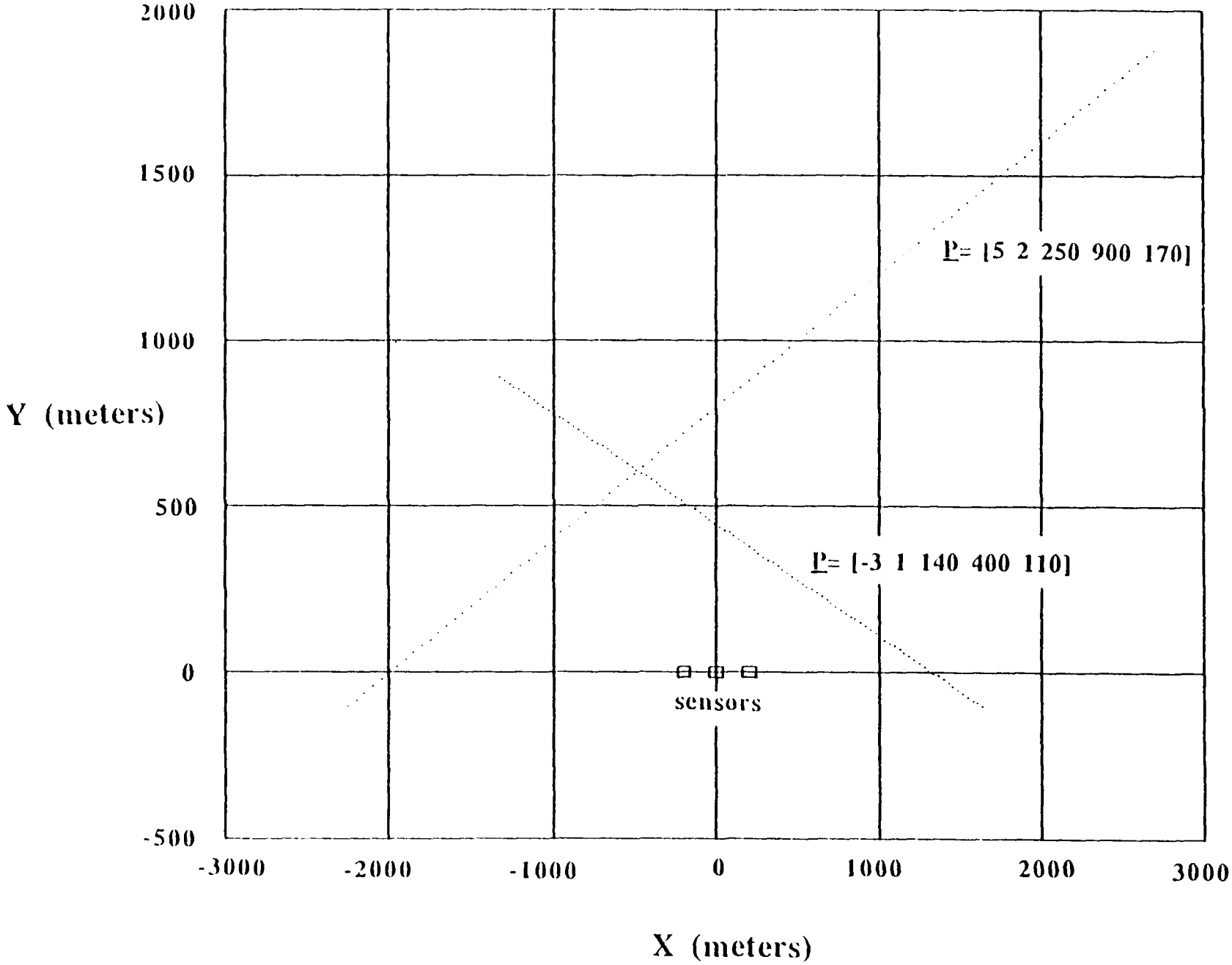

2000
1500
1000
500
0
-500
Y (meters)
-3000
-2000
-1000
0
1000
2000
3000
X (meters)
P= [15 2 250 900 170]
P= [-3 1 140 400 110]
sensors

Table 1
Monte-Carlo Runs

Run Number	Track Number	Noise Levels (meters) σ_d	$\sigma_v t$
1	1	0	10
2	1	0	100
3	1	0.1	0
4	1	1.0	0
5	1	0.1	10
6	1	1.0	10
7	1	0.1	100
8	1	1.0	100
9	2	0.1	10
10	2	1.0	10
11	2	0.1	100
12	2	1.0	100
13	1	0.1	--
14	1	1.0	--
15	2	0.1	--
16	2	1.0	--

Track parameter set #1: $[v_x \; v_y \; x_T \; y_T \; z_T]=[5 \; 2 \; 250 \; 900 \; 170]$
Track parameter set #2: $[v_x \; v_y \; x_T \; y_T \; z_T]=[-3 \; 1 \; 140 \; 400 \; 110]$

Single Sensor Location: (0,0,300)
Vertical Array Sensor Locations: (0,0,100), (0,0,300)
Horizontal Array Sensor Locations: (200,0,300), (-200,0,300)

Table 2a
Track Parameter Estimate Bias,
Single-Sensor Array

Run #	Bias (meters)			
	$\hat{x}_T - x_T$	$\hat{y}_T - y_T$	$\hat{z}_T - z_T$	$(\hat{v} - v)t$
1	-0.058	-0.090	-0.018	-0.56
2	-0.58	-0.93	-0.18	-5.6
3	-0.030	-0.084	-0.009	0
4	-0.92	-1.8	-0.15	0

Table 2b
Track Parameter Sample Deviation,
Single-Sensor Array

Run #	Sample Deviation (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
1	1.0	1.6	0.32	10
2	10	16	3.2	100
3	0.30	0.48	0.60	0
4	3.0	4.8	6.0	0

Table 2c
Track Parameter Theoretical Deviation,
Single-Sensor Array

Run #	Theoretical Deviation (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
1	1.0	1.6	0.32	10
2	10	16	3.2	100
3	0.32	0.46	0.60	0
4	3.2	4.6	6.0	0

Table 2d
Track Parameter RMS Error,
Single-Sensor Array

Run #	RMS Error (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
1	1.0	1.6	0.32	10
2	10	16	3.2	100
3	0.30	0.48	0.60	0
4	3.2	5.1	6.1	0

Table 3a
Track Parameter RMS Error,
Single-Sensor Array

Run #	RMS Error (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
5	1.1	1.7	0.32	10
6	3.5	5.5	0.70	10
7	10	16	3.2	100
8	11	17	3.2	100
9	0.20	2.0	0.35	10
10	1.9	4.1	0.52	10
11	0.36	20	3.5	100
12	2.0	21	3.5	100

Table 3b
Track Parameter Theoretical Deviation,
Single-Sensor Array

Run #	Theoretical Deviation (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
5	1.3	2.0	0.37	10
6	4.2	6.2	0.91	10
7	11	17	3.2	100
8	14	20	3.8	100
9	0.22	2.3	0.38	10
10	1.9	4.9	0.69	10
11	0.47	20	3.5	100
12	2.2	23	3.8	100

Table 3c
Track Parameter Cramer-Rao Bound,,
Single-Sensor Array

Run #	CRB Deviation (meters)			
	σ_x	σ_y	σ_z	σ_{vt}
5	1.1	1.6	0.32	10
6	3.0	4.1	0.59	10
7	10	16	3.2	99.8
8	11	16	3.2	100
9	0.17	2.0	0.35	10
10	1.7	3.3	0.47	10
11	0.36	20	3.4	97.2
12	1.7	20	3.5	100

Table 4a
Track Parameter Estimate
Bias and Standard Deviation,
Two-Sensor Vertical Array

Run	Bias (meters)				Standard Deviation (meters)			
#	$\hat{x}_T - x_T$	$\hat{y}_T - y_T$	$\hat{z}_T - z_T$	$(\hat{v}-v)t$	σ_x	σ_y	σ_z	σ_{vt}
13	0.099	0.10	0.024	0.8	0.60	0.67	0.12	4.2
14	4.5	6.3	1.4	47	5.5	5.5	0.97	34
15	0.018	0.017	0.0041	0.1	0.22	0.32	0.046	1.4
16	0.21	0.15	0.13	4.6	2.2	3.1	0.46	14

Table 4b
Track Parameter Estimate
RMS Error and CRB,
Two-Sensor Vertical Array

Run	RMS Error (meters)				CRB (meters)			
#	σ_x	σ_y	σ_z	σ_{vt}	σ_x	σ_y	σ_z	σ_{vt}
13	0.61	0.68	0.13	4.3	0.34	0.31	0.053	2.2
14	7.1	8.4	1.7	58	3.4	3.1	0.53	22
15	0.22	0.32	0.046	.14	0.15	0.14	0.020	0.96
16	2.2	3.1	0.47	15	1.5	1.4	0.20	9.6

Table 5a
Track Parameter Estimate Bias,
Two-Sensor Horizontal Array

Run #	Bias (meters)				
	$\hat{x}_T - x_T$	$\hat{y}_T - y_T$	$\hat{z}_T - z_T$	$(\hat{v}_X - v_X)t$	$(\hat{v}_Y - v_Y)t$
13	-0.11	-0.61	-0.10	2.9	-1.0
14	0.49	1.2	0.35	13	7.6
15	-0.49	-1.4	-0.25	6.7	-2.3
16	0.26	1.2	0.45	-14	5.3

Table 5b
Track Parameter Standard Deviation,
Two-Sensor Horizontal Array

Run #	Standard Deviation (meters)				
	σ_x	σ_y	σ_z	$\sigma_{v_X t}$	$\sigma_{v_Y t}$
13	1.1	4.9	0.82	24	9.9
14	10	49	8.2	240	99
15	1.3	3.5	0.65	18	6.2
16	14	40	7.2	200	68

Table 5c
Track Parameter RMS Error,
Two-Sensor Horizontal Array

Run #	RMS Error (meters)				
	σ_x	σ_y	σ_z	$\sigma_{v_X t}$	$\sigma_{v_Y t}$
13	1.1	4.9	0.83	25	10
14	11	49	8.2	240	99
15	1.3	3.8	0.70	19	6.6
16	14	40	7.2	200	68

Table 5d
Track Parameter CRB,
Two-Sensor Horizontal Array

Run #	CRB Deviation (meters)				
	σ_x	σ_y	σ_z	$\sigma_{v_X t}$	$\sigma_{v_Y t}$
13	0.24	0.90	0.15	4.6	1.8
14	2.4	9.0	1.5	46	18
15	0.086	0.38	0.065	1.8	0.53
16	0.86	3.8	0.65	18	5.3

Appendix G - A Flexible Sonar Signal Processing Workstation

WHITE PAPER

A FLEXIBLE SONAR SIGNAL PROCESSING WORKSTATION

January 1986

Prepared by:

Julius O. Smith
Yair Barniv

Approved by:

Yair Barniv
Manager,
Adaptive Systems Department

SUMMARY

A sonar signal-processing workstation is described which can be quickly reconfigured on site to provide maximum responsiveness to new and changing threats. The areas of new technology are

- A high-level programming environment tailored to digital signal processing development and implementation;
- Sophisticated software tools specifically suited to underwater detection, localization, and characterization.

The workstation thus serves in two very distinct ways, first as the ideal algorithm development environment, and secondly as implementor of "production algorithms."

Algorithms are developed interactively in a very advanced signal processing language by a highly trained signal processing expert. A successful high-level description can then be "compiled" directly into numerous low-level machine-code modules which can control a diverse installation of signal processing hardware in real time.

The workstation is simultaneously capable of high-speed development when adjusting to new threats and high-speed performance when implementing the signal processing tools thus developed.

1. INTRODUCTION

Many systems have been built to exploit the latest and most effective techniques for signal processing in the context of underwater surveillance. Software development for these systems has typically relied on the tools provided by computer manufacturers. Once a system is developed for a particular task, it is normally not easy to modify in the field on a fundamental level, and the user may be forced to choose among a variety of menu selections, unable to contribute fundamental enhancements to the system. While appropriate for lowly skilled operators, such systems cannot be developed on site, thereby significantly under-utilizing the experience and insight of the highly trained operator. The system proposed in this paper can close the gulf between designer and end-user; fundamentally new approaches to target detection can be installed literally overnight.

This paper outlines a workstation concept one level higher than the typical: In addition to providing a powerful set of tools for acoustic underwater surveillance, we will provide a system maximally effective for developing future surveillance systems. This allows maximum flexibility with respect to countermeasures which may render existing systems ineffective.

2. FEATURES OF THE SIGNAL PROCESSING DEVELOPMENT SYSTEM

Present-day software tools for maximally advanced development consist of the following features:

Interactive high-level language (with compiler)
Integrated editor
Self-documenting and self-checking applications tools

2.1 HIGH-LEVEL SIGNAL PROCESSING LANGUAGE

Modern software development systems include an interactive programming language which supports the notation and needs of the application to the greatest possible extent. In the case of underwater acoustic signal processing, this language would support complex matrix operations, polynomial manipulations, controllable numerical precision, and operations applied to an indefinitely long sequence, to name a few. The language must also support the development of concise mathematical notation as close as possible to that used by a design engineer working with paper and pencil. The nearly mathematical descriptions manipulated by the design engineer are executed interpretively in the development stage, allowing maximally high-level debugging and evaluation, but the final equations are automatically translated into ultra-fast machine instructions by a hard working compiler.

Unfortunately, there is no language for signal processing which achieves a satisfactory percentage of the above goals. The principal language in current use for signal processing is Fortran, although in the future, ADA will be extensively used. While proposed changes to Fortran over the next decade may give some support to treating arrays and even matrices as primitive objects in the language, there is no real possibility of obtaining full support of signal processing needs from either Fortran or ADA in the foreseeable future.

The best obtainable signal processing language to date appears to be CtrlC (pronounced "control-see"), a software product of Systems Control Technology (SCT). CtrlC is an extension of MatLab (for "matrix laboratory"),

which is a numerical analyst's desk calculator language invented by Cleve Moler at the University of New Mexico. MatLab expressions involve a single data type: matrices of double precision complex floating-point numbers. MatLab expressions are extremely concise for linear algebraic equations. For example, typing "x=b\A" in MatLab solves the matrix equation Ax=b using Householder reductions followed by back-substitution (Dr. Moler is a leader in the field of matrix numerical analysis and was involved in the development of EISPAK for eigen analysis). If Ax=b is overdetermined, then a a numerically robust version of the least squares solution $x = (A^H A)^{-1} A^H b$ is produced. MatLab contributed fundamental progress in the development of a high-level syntax for digital signal processing operations.

CtrlC is the only extension of MatLab which supports user-defined functions. This extension is critical for the use of CtrlC as a high level signal processing language. The design engineer can very quickly build up a large library of functions written in CtrlC. Compared with writing Fortran, the CtrlC environment is about one order of magnitude faster for algorithm development. This advantage is due to the high-level syntax and the fact that it is interactive. To date, there is not yet a compiler for the CtrlC language. Automatic translation from CtrlC to Fortran is one subtask of the workstation development project.

2.2 INTEGRATED EDITOR

In addition to the language itself, a modern software development system provides an "integrated editor" which "understands" the high-level language, and contains tools fitted to the same application. For example, a signal processing language editor would provide extensive display support for digital signals and system diagrams. The editor should support both text and graphics as program source. Text is used to describe mathematical functions in a concise yet maximally clear manner, and block diagrams are used to describe whole systems. (Block diagram notation is especially important when making extensive use of parallel computation.) Diagram blocks can include text which contains, for example, the mathematical expression for the function of that box, or the name of a signal processing primitive and a pointer to the list of parameter values for that instantiation of it. (Signal processing primitives

very often have quite a few parameters which must be fixed by the designer. It is normally necessary to effectively provide a table of parameters for each processing box. They are easily managed via menu-style interfaces.)

In a well-integrated editor, a high level "expression" (mathematical syntax or a piece of a system diagram) can be "evaluated" in the editor's environment to produce high level objects which can be inspected or used in subsequent operations. During system development, it is never necessary to leave the editor. All results of all operations exist in forms which can be further manipulated, displayed, or transformed within the editor. Later, when the processing functions have been identified, and a complete system has been configured, the collected functions and/or diagrams are then compiled by another editor command, and the net result is a high-speed version of that capability. The compiled code is then available as a (fast) primitive in the editor. It is also available for installation as an operator's option in a menu tree, or as a permanent ongoing real time process (such as needed for automatic detection of a new threat).

SCT is presently committed to the development of the block-diagram interface to CtrlC. Development of an integrated editor environment "around" this facility and the CtrlC language is a subtask of the proposed workstation development project.

2.3 SIGNAL PROCESSING TOOLS

In addition to the language and an integrated editor, there should be an extensive library of sonar signal processing tools which support the design engineer in every conceivable way. For example, primitive functions should exist for computing the FFT, power spectral density, autocorrelation function, cross-correlation, Doppler correlation, target localization from delay or Doppler information, digital filter design, sampling-rate conversion, coherent and non-coherent sinusoidal line tracking, peak finders, statistical utilities, signal generators for testing, image processing utilities, and so on. These primitives introduce higher level data types which can be given "methods" describing how to display them, or even how to interpret them in high-level expressions.

All of the signal processing primitives above and more have been developed in Fortran and CtrlC at SCT. Due to lack of a wide customer base, only the most general signal processing utilities are being sold in the commercial distribution of CtrlC. With a relatively small level of effort, a complete installation of sonar signal processing tools into CtrlC can be carried out at SCT. This alone would seem to provide the most advanced sonar-signal-processing detection workstation in the world.

Part of the process of integrating signal processing tools into CtrlC is to supply intelligent defaulting of unspecified arguments, and consistency checking for the arguments. Here the algorithm designer can save the user from simple mistakes such as undersampling (which produces aliasing) and "do the right thing" when various optional parameters are omitted in the interactive calling sequence. Hardening of the user interface in this way is expensive but very valuable. It and the function documentation (described below) will account for nearly all of the time it will take to integrate existing SCT acoustic surveillance tools into CtrlC.

2.4 SELF-DOCUMENTATION

Finally, given a powerful language and integrated editor, together with a comprehensive set of primitive signal processing functions, there must be a very extensive online documentation facility. This documentation should allow the expert engineer to sit down and get to work immediately without reading any manuals and with a minimum of "database queries" on his part. Fast acquaintance with the enormous library of instantly available primitives is made possible by "keyword searching" through the function-names, or any of the various levels of function documentation described below. If he knows the name of the function he wants (or the first few characters of it), he can ask for a display of the documentation for that function at one or more levels. He can be in middle of typing a call to a function in an expression and request information (such as a description of the function's arguments) to be displayed in a side window without having to leave the expression he is typing.

Each primitive function should be documented on several levels: (1) a

one-line description of what the function does, (2) a list of argument mnemonics and their default values, (3) a list of one-line descriptions of the function arguments, (4) a full description of the function's purpose and useage, (5) high-level source code for the function, and (6) application notes and examples. Cross-references can be generated to pre-existing systems using any given function (to allow inspection of parameters for comparable situations in the past).

2.5 PROJECTED BENEFIT

In conclusion, development of the high-level workstation we propose promises the following benefits:

- Subcontractor signal processing development will produce better results per dollar by an order of magnitude or more.
- The time required to configure a new signal-processing system (made necessary, for example, by the need to track new types of targets), will be reduced by at least an order of magnitude.

These benefits are achieved by providing the best existing software development tools, uncompromising high-speed implementation, and a complete battery of all relevant signal processing techniques.

3. SCT BACKGROUND

The need for the advanced development environment discussed here has become increasingly apparent after many man-years of signal processing software development at SCT. SCT has accumulated an extensive capability in signal processing and has devoted special emphasis over the past several years to passive underwater surveillance systems. In addition to a full complement of sonar signal processing utilities (many implemented on an FPS array processor), SCT has developed an extremely powerful dynamic programming approach to multitarget tracking. In related projects, extensive capabilities in digital image processing have been developed over the years.

The comprehensive interactive programming environment will allow rapid prototyping of signal processing systems on the basis of experiments with realistic data. CtrlC, developed by SCT, provides an initial start in this direction. While users of CtrlC are extrememly happy with it, the next implementation will be far more powerful.

In the following two subsections, some existing signal processing utilities and some CtrlC capabilities are listed.

3.1 EXISTING SONAR SIGNAL PROCESSING TOOLS AT SCT

For the past several years the Adaptive Systems Department of SCT has had ongoing contracts with the Navy and others to develop new digital signal processing techniques, mainly in the area of underwater surveillance. The emphasis has been on spectrum analysis and source localization from multiple hydrophone recordings. As a result, the following SOFTWARE capabilities now exist:

- Power spectral density estimation using the FPS array-processor (AP)
- Estimation of crosscorrelation and autocorrelation functions on the FPS AP
- SCOT and PHAT normalized cross-spectra and correlati on FPS AP
- Synthetic FFT on the FPS for narrowband "zoom" (The FPS maximum is 8K)
- Arbitrary FFT window generator
- General digital filtering on the FPS AP

- Sampling rate conversion on the FPS AP
- Digital filter design software to support the above
- Modified Yule Walker "high resolution" spectrum analysis
- Doppler cross-correlation (a broadband tracking technique)
- A multitarget localization program
- Adaptive tracking of multipath delay and multipath reflection coefficient
- Adaptive delay estimation
- Synthetic signal generation (1 or 2 sinusoids in white or filtered noise), Gaussian noise generation, chirp sinusoid, impulse
- Spectral noise-floor equalizers
- Synthetic Doppler generator
- Signal statistics measurement
- Signal interpolation
- The Constant-Modulus Algorithm for multipath estimation and channel equalization
- Recursive maximum likelihood method for adaptive filtering and estimation
- Overdetermined Instrumental Variables method for adaptive filtering
- Maximum A Posteriori Line Extraction (MAPLE) for coherent narrowband case
- ADEC for tracking noncoherent sinusoids in noise
- IEEE Programs for Digital Signal Processing
- EISPACK, LINPACK matrix analysis packages (e.g. least squares, eigenvalues)
- Graylevel 2D display (for spectrograms, correlograms, and the like)
- Graylevel output drivers for the HP LaserJet, Printronix, and DeAnza.
- Calibrated line-track display suitable for overlay on graylevel plot
- Utilities for reading and unblocking sonar data received on magnetic tape
- Support utilities for data IO in several formats.

Conversion programs exist for going between any pair of formats below:

16-bit packed integers (typical for raw data)
32-bit floating-point (typical for intermediate data)
32-bit complex floating-point (used for some intermediate data)
CtrlC binary format (can load any data into CtrlC for examination)
MassLib format (SCTLIB image-processing tools require this format)

All of the above programs are written in Fortran or CtrlC and currently run on the Vax 11/785. In all, we have several hundred more general-purpose signal processing utilities which combine to give a complete and well-rounded set of tools. It is high time for a new phase aimed at integrating these tools in a manner that will put them effectively in the hands of the capable user.

3.2 SUMMARY OF CURRENT CtrlC FEATURES

CtrlC is currently aimed primarily at the control design and general signal processing community. A feel for the present CtrlC environment is gleaned by typing HELP:

```
$ DO CTRLC
[> help
                              Ctrl-C Help Facility
Starting & Ending          Examples              Commands Summaries
Basics                     Graphics & Plotting   Procedures & Functions
Files & Data               Matrix Computations   Control Systems
Filtering & Estimation     Modeling & Simulation Signal Processing
Identification             Numerics & Utilities  Personalizing Ctrl-C

      o Type help and the first unique letters of the topic for info
      o Type help and the first letters of ANY topic to check for info
      o If you are new to Ctrl-C, type 'help intro'
      o For more information on how to use the Help Facility, type
            'help help'
      o More detailed information is available in your Ctrl-C manual
```

Appendix A contains a complete list of the functions presently delivered with CtrlC.

4. AN EXAMPLE SCENARIO

In this section we try to convey a "feel" for the proposed system by describing an imagined scenario in which an outside consultant has been called in to develop a new processing technique using the proposed signal processing development system. To bring out as many novel aspects of the system as possible, we assume the consultant has no prior experience with it. He has been told only that he can push the 'help' button if he has any questions about the system or what it can do.

The Scenario

An underwater acoustics expert has been asked to take a look at some unidentified underwater signals which have been recorded by a sensor array. It is thought that these sounds are emanating from a new type of underwater vehicle. The objectives are to infer the power spectrum of this new source (as a function of its orientation, range, velocity, and acceleration), and to develop software for detection, localization, and tracking in real time.

The consultant sits down at the console and presses the help' key. An "editor window" is formed containing a summary of further help topics. (This is characteristic of the way all requests for information are handled: a new editor window is created initially filled with the requested information.) The help topic summary might look as follows:

Select any of the following:

* INTRO - Orientation for the new user
* HELP - How to use the HELP facility
* LANGUAGE - High level signal processing language tutorial/summary
* TOOLS - Heirarchical synopsis of available signal processing tools
* SYSTEM - Documentation of system utilities (editor, windows, etc.)

(Select by pointing to desired line with mouse and pressing mouse button, or type capitalized keyword above followed by the RETURN key.)

The user then "clicks the mouse" on the INTRO line (or types INTRO <RETURN>). The introduction provides a general overview of the system and gives enough initial information to allow the user to proceed with the other initial help' choices.

The format of each help topic is hierarchical. The goal is to make it possible for a new user to obtain any level of detail in a logical manner, or allow an experienced user to get a terse summary of features with one or two mouse clicks. The menus at each level of the tree are arranged so that the experienced user gets to the most often needed summary windows as fast as possible (e.g. the mouse does not have to be moved). In all cases the help information appears in a separate window away from the "working window" in which the user is developing software.

The first time user will want to peruse the documentation for at least an hour or so, skimming through much of the tutorial level information and studying the terse summaries in more detail. The signal processing language will take the most study time, although it will be very natural to use (being closer to true mathematical notation than typical programming languages and having less general data structures and data types to learn about). The language tutorial will include examples and "worksheets" where the new user can practice language features. The TOOLS help subtree is organized into categories, much as the initial help display of CtrlC is currently presented (see above). All told, it should not take more than one day to become fully familiar with the general capabilities of the system and how to find specific information when needed.

So, to work. Our acoustics expert wants first to see power spectrum estimates from the three nearest sensors over time intervals centered about CPA (closest point of approach). He therefore types

?spect<cr>

where <cr> denotes the RETURN key, and a window appears containing a list of all tools with the word fragment 'spect' in their titles. Similar commands create a list of tools with 'spect' in their one-line summaries, argument

documentations, or full documentations. In this way the user can control the depth of the keyword searching. In addition to keywords, he can specify so-called "regular expressions" for the search key. Regular expressions are a notation for complex pattern matching used in sophisticated text editors.

The function 'spectrogram' looks promising, so he clicks the mouse on that line of the help window, or he simply types

spectrogram^?

where ^? denotes <CONTROL>?. A function name followed by ^? is a request for a help window on that function. At any time a quick summary of the various '?' options can be created by pressing the <HELP> key, selecting HELP, and then selecting HELP SUMMARY. This window can be left open in the upper right-hand corner of the screen for constant future reference if desired. The screen is large enough for several windows to be open at once. Windows can be shuffled around and resized at will. They can even be "buried" and later "revived" like papers on a desk.

The next task is to apply the spectrogram function to each of the three closest sensors. Suppose the passive sonar data for 10 channels exist in files Cn.DAT, for n=1,2,...,10. The time of CPA in each channel is estimated by typing (notice that the text to the right of // represents comments)

```
EnergyEnv(x,span[5]) := {          // Define new function
// Return energy envelope = unnormalized moving average of x(i)**2
  xs=x.*x;                         // Array of x(i)**2
  FOR i=1:LENGTH(x),..             // Lowpass filter (LENGTH is intrinsic)
    y(i)=SUM(xs(i-span+1:i));      // xs(i) = 0 for i<1. (SUM is intrinsic)
  RETURN(y);                       // Signal envelope estimate
}

FOR i=1:10, [e(i),cpa(i)]=MAX(EnergyEnv(X(:,i)=GetSig("C"&i2s(i))));
```

The above is equivalent (except for the optimal argument 'span' whose default value is 5) to the following Fortran:

```
      SUBROUTINE ENERGYENV(Y,X,N,ISPAN,WORK)
      DIMENSION Y(N),X(N),WORK(N)
      DO 10 I=1,N
      WORK(I) = X(I)**2
10    CONTINUE
      DO 20 I=1,N
      K=I
      SUM=WORK(K)
      DO 30 J=1,ISPAN-1
      K = K - 1
      IF (K.LT.1) GO TO 40
      SUM = SUM + WORK(K)
30    CONTINUE
40    Y(I) = SUM
20    CONTINUE
      RETURN
      END
C
      PARAMETER NMAX = 10000
      DIMENSION E(NMAX),ICPA(NMAX)
      DIMENSION TEMP1(NMAX),TEMP2(NMAX),TEMP3(NMAX),X(NMAX,10)
      NCHANS=10
      DO 10 I=1,NCHANS
      CALL GETSIG(N,TEMP1,MAKENAME('C',I),NMAX)
      DO 15 J=1,N
      X(J,I) = TEMP1(J)
15    CONTINUE
      CALL ENERGYENV(TEMP2,TEMP1,10,5,TEMP3)
      EMAX = TEMP2(1)
      MAXLOC = 1
      IF(N.LE.1) GO TO 30
      DO 20 J=2,N
      IF(EMAX.GE.TEMP2(J)) GO TO 20
      EMAX = TEMP2(J)
      MAXLOC = J
20    CONTINUE
```

```
30      E(I) = EMACS
        ICPA(I) = MAXLOC
10      CONTINUE
        END
```

Clearly, Fortran is not nearly as expressive as the (slightly extended) CtrlC syntax above. In addition to eliminating manual handling of loop counters in many situations, the CtrlC language provides for any size of input whereas the Fortran version must predefine a maximum array size (and actually allocate that much core memory whether or not it is ever used).

If a Fortran version is needed (e.g. for portability or compilation on a fast processor), the CtrlC code can be automatically translated into Fortran by typing, for example,

```
Fp = FORT(EnergyEnv);       // Fp is a string containing the Fortran
DISP(Fp);                   // Print the Fortran on the terminal
SAVE Fp >FortSubs.F77 -append  // Write to disk
```

If the above is being done often, a simple function 'SaveF77' can be quickly defined so that the command 'SaveF77(EnergyEnv)' will accomplish the same result.

The experienced user would have the function 'EnergyEnv' already defined, and finding the time of CPA would involve only one line of CtrlC. The user would have specified sometime earlier (typically in his automatic startup options) that the file 'FortSubs.Olb' should be searched for any undefined functions. 'FortSubs.Olb' is library of COMPILED Fortran modules with a directory of entry points at the beginning. The directory allows random access within the file for fast selective loading as needed. After the directory has been searched the first time, the directory is retained in core to speed up any future searches for unrecognized functions.

The three closest sensors are now found by typing

```
key = 1:10;                          // Passive sort key
[Ed,CPAd,key] = SORT(E,CPA,key);     // Ed = E in descending order.
                                     // Remaining args sorted passively.
Closest = key(1:3);                  // Take three closest
```

The channel numbers of the closest three sensors (as measured by maximum signal energy at CPA) reside in the length three array 'Closest'. A spectrogram of the closest sensor is obtained as follows:

```
S1 = Spectrogram(X(:,Closest(1)));   // Take numerous default parameters
Show(S1);                            // Display spectrogram (2D gray or 3D)
```

The default spectrogram parameters provide a standard "Lofargram" which is a smoothed short-time power-spectral-density estimate (power density versus time and frequency). The default arguments can be overridden in either a "sticky" or "nonsticky" fashion. Sticky overrides become the defaults in future calls to the function. A list of all of the arguments to Spectrogram and their defaults is obtained in a side window by typing "Spectrogram(^?" (<CONTROL>? inside a function call). The list of defaults can be edited right there in the documentation window, and a command exists which installs the edited argument information as new defaults. Nonsticky defaults are listed to the right of the original defaults. Sticky defaults are obtained by typing over the old default.

Having looked at the spectrograms, our consultant has determined that there is very little narrowband information associated with the source, but that there is a lowpass broadband "shelf" extending to about 800 Hz. By comparing the measured spectrograms against "noise cuts" one hour earlier (again quickly accomplished interactively in a few minutes), he has determined that there is no useful information above 2KHz. Therefore, it will be good to lowpass filter and reduce the sampling rate of the sensor channels down to 5KHz or so:

```
for i=1:10, XR(:,i) = SrConv(X(:,i),5000/Fs); // Fs = original sampling rate
F_s=5000'
X=Xr;
CLEAR(Xr); // Free up memory
```

Now the CPA times will be used to form an estimate of the power spectrum. First, however, we plot the spectra at CPA in an overlay to check for bad data (due to interference, low signal level, faulty sensor, or the like):

```
FOR i=1:10, S(:,i) = ABS(FFT(Hamming(Xr(Cpa(i)-512:Cpa(i)+511))))^2
S(:,11) = NoiseFloor; // Previously derived noise floor estimate
Show(S,'Overlay');    // All columns of matrix S plotted in overlay fashion.
```

Actually, it is more efficient to say "S(:,i)=(t=FFT(...)).*CONJG(t)" instead of "S(:,i)=ABS(FFT(...))^2 ," but that's a fine point. Note that each spectrum can be displayed using a different color in the overlay. Color coding makes it possible to follow individual spectra in the overlay more easily.

From the spectral overlay, it is determined that only the closest seven sensors have worthwhile measurements. We wish to average these to produce a single estimate of the source spectrum. However, it is not ideal to simply average because some sensors have better looking spectra than others. A weight function is formed from the energies at CPA as follows:

```
W = Ed(1:7);          // Weighting is proportional to energy at CPA
W = W/SUM(W);         // Normalize
```

The estimate of the power spectrum is given by a weighted average of the spectrum at CPA in each of the seven sensors closest to the source:

```
Starget = S(:,key(1:7))*W;        // Yes, vectors can be used as indices
```

The estimated target PSD can be hand edited in the waveform editor to remove interference. Also it may be possible to extrapolate the source spectrum in a physically plausible manner past the frequency where it dips

below the noise floor. For example, if the spectrum is rolling off at 12dB per octave when it merges with the background noise, it is reasonable to extrapolate the 12dB per octave rolloff.

The consultant was mindful that the above process of estimating a power spectrum by averaging CPA spectra might occur again. Therefore, near the beginning of his session he started up a "history window," and every keystroke typed is being saved in this window. The contents of the history window can be written out to a disk file at any time and/or used as a "batch file" later on. At any time he can "mouse over" to the history window and make edit changes (to eliminate a typo or digression, for example), or he may even grab a set of earlier commands and execute them. At all times the full power of the editor is available in every window.

Now we have a preliminary estimate of the source power spectrum. Using it and the noise cut, we can estimate a Wiener filter to be used in making broadband differential Doppler measurements:

```
Filt = Wiener(Starget,NoiseFloor);
```

In reality a separate Wiener filter would be desired for each time step in the short-time spectra. Also it may be desirable to edit this weight function in the waveform editor to downweight frequency bands of high "variance" in the PSD estimate. That is, the Wiener filter only penalizes closeness to the noise floor while in reality there should also be a penalty for nonstationarity (or "spuriousness") over time. The spectral "reliability measure" can be computed quantitatively or drawn according to the judgment of the signal processing engineer. Whatever the ultimate weighting, let Filt' contain the final spectral window.

Next we attempt to localize the target by estimating differential Doppler tracks in the seven good sensor recordings. Differential Doppler is measured by cross-correlating the Weiner and confidence-weighted spectra, after resampling the frequency axis on a log scale. The details are similar to the above examples and will appear in a forthcoming final report on a project of this nature. (The work was fully carried out within CtrlC). Suffice it to

note that development of broadband Doppler correlation software was greatly facilitated by the high-level interactive programming environment provided by the present version of CtrlC. The planned extensions to CtrlC will take it far beyond the present capabilities.

After obtaining differential Dopplers plus, perhaps, differential time-of-arrivals (from temporal cross-correlations) and multipath measurements, along with their associated weight functions, the data are passed to a comprehensive multitarget tracking facility (based on dynamic programming as discussed in the next section). The results of localization can then be used to deconvolve the propagation channel and obtain calibrated spectral estimates as a function of orientation and all operating parameters of the target.

The proposed system has been described from the point of view of a new user trying to develop algorithms almost from scratch. This aspect of the system has been stressed because it is the most unusual feature: algorithm development is supported at the highest possible level.

To translate a finished algorithm design into a deployable form, it is only necessary to "clean up" the code in the history window, translate various CtrlC functions into Fortran (or other suitable "intermediate code"), compile the Fortran for the target machine (such as an array processoror "super computer" configuration), and build a block diagram of the final processing system. Each block in the diagram contains the name of the function that goes into it. Block diagrams are not essential, but they are convenient descriptions of complex, parrallel, real-time processing tasks. They also serve better as "prints" for the final system. Using "software probes" (controlled by the mouse) windows can be formed which display the waveform at any point in the diagram (while it is executing in real time), or values can be printed into a side window and further processed in an unrestricted manner much as described above. Several windows can be set up to display waveforms at several different points within the network. A waveform display window can also be allocated to automatically display the results of each line of interactive computation in the CtrlC language. At any time the user may move over to the waveform and interactively manipulate it.

This discussion has only scratched the surface of the capabilities of the proposed system. Needless to say, a considerable effort will be required to fully implement the concept. However, the commercial availability of good interactive programming environments, together with SCT's experience and progress to date in this area, render the project feasible.

References for the High-Level Signal-Processing Workstation Concept

[1] T. J. Aird, "The Fortran Converter User's Guide", IMSL, 1975.

[2] D. R. Barstow, H. E. Shrobe, and E. Sandewall, eds., "Interactive Programming Environments", McGraw-Hill, New York, 1984.

[3] J. Boyle and K. Dritz, "An Automated Programming System to Aid the Development of Quality Mathematical Software", IFIP Proceedings, pp. 542-546, North-Holland Publishing Co., Amsterdam, The Netherlands, 1974 .

[4] W. J. Cody, "The FUNPACK Package of Special Function Subroutines", ACM Trans. Math. Software, vol 1, pp 13-25, 1975.

[5] W. Cowell, "Portability of Numerical Software (Lecture Notes, Computer Science, vol. 57)" Springer-Verlag, New York, 1977.

[6] W. Cowell and L. J. Orsterweil, "The Toolpack/IST Programming Environment", Argonne National Lab., Applied Math Div., Report ANL/MCS-TM7, 1983.

[7] Digital Signal Processing Committee, ed., "Programs for Digital Signal Processing", IEEE Press, New York, 1979.

[8] J. Dongarra et al., "LINPACK User's Guide", SIAM, Philadelphia, PA, 1979.

[9] G. E. Forsythe and C. B. Moler, "Computer Methods for Mathematical Computat ions" CS Dept., Stanford Univ., Dec. 1972.

[10] L. D. Fosdick, Ed., "Performance Evaluation of Numerical Software", North-Holland Publishing Co., Amsterdam, The Netherlands, 1979.

[11] B. S. Garbow, et al., "Matrix Eigensystem Routines--EISPACK Guide Extension (Lecture Notes in Computer Science, vol. 51" Springer-Verlag, New York, 197 7.

[12] M. A. Hennel and L. M. Delves, Eds., "Production and Assessment of Numerica l Software", Academic Press, New York, 1980.

[13] K. Madsen and J. K. Reid, "Fortran Subroutines for Finding Polynomial Zeros ", Report 7986, Computer Science and Systems Division, A.E.R.E. Harwell, Didcot, Oxford England, Feb. 1975.

[14] P. Messina and A. Murli, Eds., "Problems and Methodologies in Mathematical Software Production, (Lecture Notes in Computer Science, vol. 142)", Springer-Verlag, New York, 1982.

[15] C. B. Moler, "MatLab User's Guide", CS Dept., Univ. of New Mexico, Nov. 1980.

[16] J. J. More, B. S. Garbow, and K. E. Hillstrom, "User Guide for MINPACK-1", Argonne National Lab., Report ANL-80-74, Aug. 1980.

[17] "The Relationship between Numerical Computation and Programming Languages," North-Holland Publishing Co., Amsterdam, The Netherlands, 1982.

[18] B. G. Ryder, "The PFORT Verifer: User's Guide", AT&T Bell Labs, CS Tech. Rep. 12, 1975; and "Software Practice and Experience" vol. 4, pp. 359-377, 1974.

[19] B. T. Smith, et al., "Matrix Eigensystem Routines--EISPACK Guide, (Lecture Notes in Computer Science, vol. 6, 2nd ed." Springer-Verlag, New York, 1976 .

[20] S. Zohar, "Fortran Subroutines for Solution of Toeplitz Sets of Linear Equations", IEEE Transactions on Acoust., Speech, and Signal Proc., vol ASSP-27, pp. 656-658, Dec. 1979. See also vol. ASSP-28, p. 601, and vol. ASSP-29, p. 1212.

[21] S. Wolfram, "Symbolic Mathematical Computation", Comm. of the ACM, vol 28, pp. 390-394, April 1985.

[22] S. Wolfram, "SMP Reference Manual", Computer Math. Group, Inference Corp., Los Angeles CA, 1983.

[23] A. Hearn, "REDUCE 2 User's Manual", Rep. UCP-19, Univ. Utah, Salt Lake City , 1973.

[24] Mathlab Group, "MACSYMA Reference Manual", Lab. for Computer Science, MIT, Cambridge MA, 1977.

[25] D. R. Barstow, H. E. Shrobe, E. Sandewall, eds., "Interactive Programming Environmnts", McGraw-Hill, 1984.

[26] Steele, Guy L., "Common Lisp, the Language", Digital Press, 1984.

[27] J. F. Reiser, ed., "SAIL" Report No. STAN-CS-76-574, Computer Science Dept., Stanford University, Aug. 1976.

[28] B. W. Kernighan and P. J. Plauger, "Software Tools", Addison-Wesley, Redding, MA, 1976.

[29] G. E. Kopec, "The Integrated Signal Processing System ISP", Fairchild Tech. Rep. no. 644, FLAIR Tech. Rep. no. 21, June 1983.

[30] D. Weinreb and D. Moon, "Lisp Machine Manual", MIT Artificial Intelligence Lab., Cambridge, MA, July 1981.

[31] J. Little, "CTRL-C: A Language for the Computer-Aided Design of Multivariable Control Systems--User's Guide", Systems Control Tech., Inc., Palo Alto, CA, 1985.

[32] ISI staff, "MATRIXx User's Manual", Integrated Systems Inc., Palo Alto, CA, 1985.

5. DYNAMIC PROGRAMMING APPROACH TO MULTITARGET DETECTION

In this section we go into more detail regarding our dynamic programming approach to target detection and localization. Essentially, sensor measurements are compared to expected measurements given the target state, sea state, propagation models, and clutter conditions. While such an approach is clearly as powerful as any conceivable for target detection and tracking, it demands extensive signal processing and modeling support such as can be provided by the proposed system.

5.1 MULTITARGET DETECTION IN A VERY-LOW-SNR ENVIRONMENT

In our recent efforts on multitarget localization we have dealt with cases where the frequency lines appearing in the Spectrograms (or other "Grams") are relatively easy to recognize--at least by the human eye. In such moderate-SNR cases it is possible to automate the line detection process using some off-the-shelf line-tracking algorithms such as ADEC or MAPLE which we have done successfully.

When the SNR of the Spectrogram data falls under some critical minimum, the above line tracking algorithms collapse. In such cases we suggest to apply a much more powerful approach which has been developed at SCT for the problem of detecting/tracking very dim airborne targets as perceived in a set of IR mosaic sensor imagery. This proprietary algorithm is based on what is referred to as Dynamic Programming Algorithm (DPA) or Viterbi Algorithm, and can be shown to solve the detection/tracking problem optimally. In the following we will describe the theory and possible implementation of this algorithm.

Although the algorithm is very general, we prefer to describe it in the context of our particular problem to be defined as follows. Having a set of sensors and the various "Grams" that can be obtained from each (such as LOFARgram, Spectrogram, Correlogram and Doppler Correlogram), detect, localize, and track targets in the optimal way so as to incorporate all the available data and side information into the solution.

For simplicity, let us assume that a Spectrogram output for each sensor is the only type of measurement available. Further, we are looking for submarines which are known to emit a single narrowband frequency and to travel at a constant velocity within some given speed window. The unknowns are (a) the number of targets (which may also be zero) in the ocean volume under surveillance. (b) the speed, location, and direction of each target.

Let us now imagine a single target travelling from point (x_1, y_1, z_1) to (x_2, y_2, z_2) during the Spectrogram time period, say 10 minutes. For this target we can predict the Spectrogram lines that should appear in each one of the sensors given their (x,y,z) coordinates. These

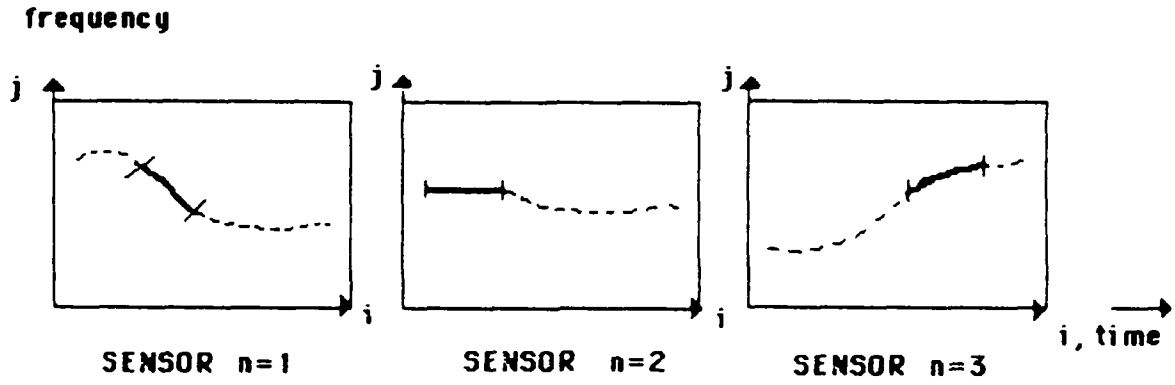


Fig. 5.1. Spectrograms for 3 Sensors

lines may look like the dashed lines shown in Figure 5.1. We can thus always calculate the expected noise-free Spectrograms for any number and location of sensors for any given target trajectory and period of time.

If we now discretize the time/frequency Spectrogram images by, say, 10 seconds/1 Hz pixels, we will be able to store the above information in a triply-indexed (3 dimensional) table. The table indices will be n for the sensor number and (i,j) for the time/frequency pixel location; its entries will be the calculated power at each pixel. In other words, we have mapped a given general target trajectory into a triply-indexed table of expected signals. The next step would be to also discretize the relevant ocean volume into cells and map all possible (straight) target trajectories

$x \triangleq (x_1, y_1, z_1, x_2, y_2, z_2)$, into tables using the procedure described above.

We can now "matched filter" (MF) the actual (discretized) Spectrogram data with the prepared bank of filters in parallel, and use their outputs to determine the likelihood of each trajectory (see Fig. 5.2). It is clear that, if no target is present, the outputs of all MFs will consist of noise only, and when a target is present the MF that "fits" the trajectory will show the maximum output. This way we will be able to detect and track simultaneously.

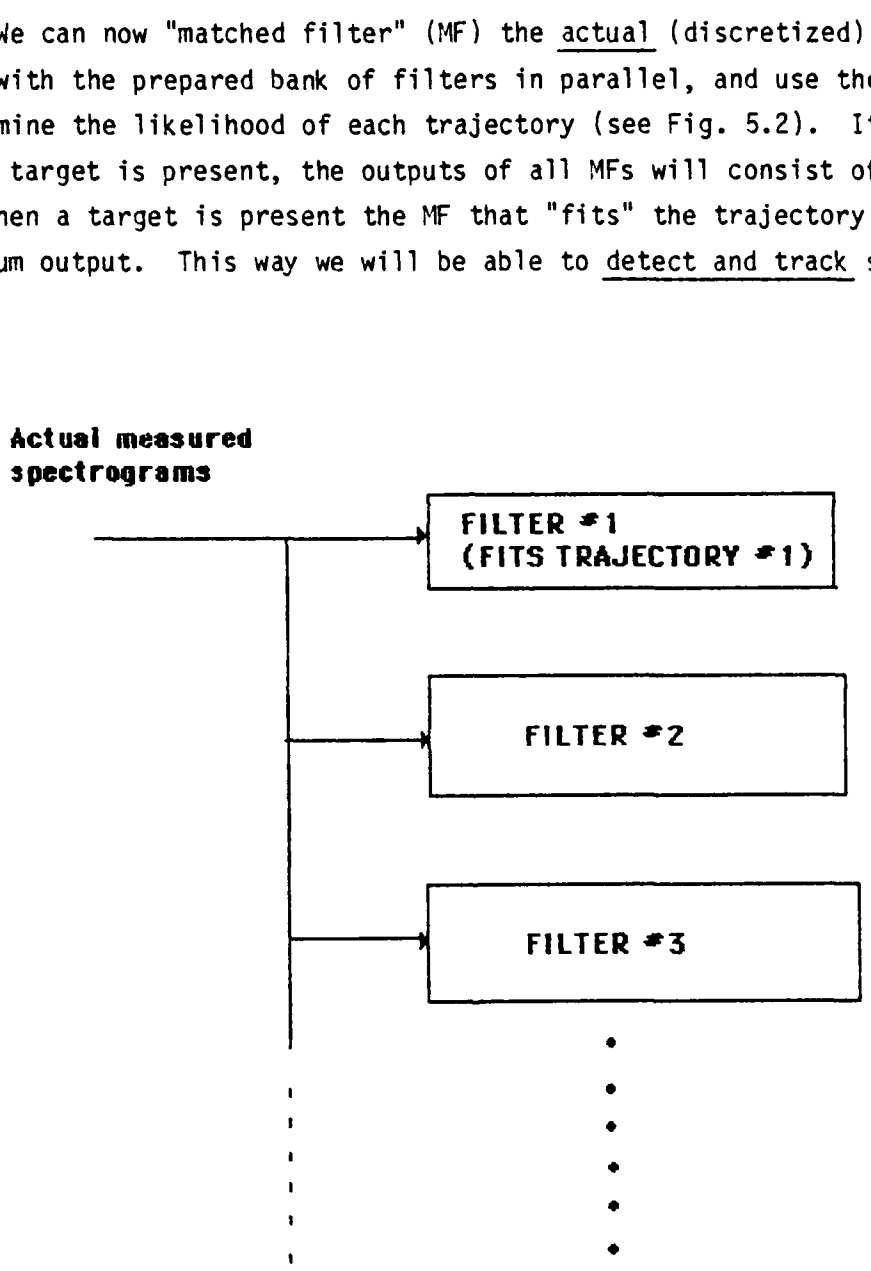


Figure 5.2. Filtering the spectrograms data through all Filters. Each Filter "fits" a Particular Straight Target Trajectory in the Ocean.

The above-described conceptual algorithm amounts to an intensive exhaustive search over all possible straight trajectories. In practice, we are of course also interested in non-straight trajectories. It is easy to realize that searching, even just for the straight trajectories, is a monumental and infeasible computational task. There is, however, a feasible way to do the equivalent of this and other exhaustive search problems; the "magic" is to use the Dynamic Programming Algorithm (DPA) to do the above search iteratively stage-by-stage which requires many orders of magnitude less computations and memory. The way we can apply the DPA here is the following.

We define all possible short (say, 1-minute) and straight trajectory segments as states, $x \triangleq (x_1, y_1, z_1, x_2, y_2, z_2)$, where the end points of each state are given by their discrete ocean-volume indices. For each such state we will calculate the table of expected Spectrogram discrete measurements for all sensors. This table will of course be much smaller than the one required to describe a complete trajectory. Next, we define the DPA stages to be 1-minute intervals, for example. A set of k consecutive states,

$$X_k \triangleq \{x_0, x_1, \ldots, x_k\} ,$$

would thus describe a complete trajectory composed of short straight segments, and, as such, would also allow curved trajectories (notice that the index k of x_k is now used as a stage number (related to time) and not as a discrete volume index).

We now denote the set of single-stage discrete Spectrogram measurements from all sensors by z_k (the solid lines in Fig. 1) and the sequence of such sets for all stages (the complete Spectrograms) by

$$Z_k \triangleq \{z_1, z_2, \ldots, z_k\} .$$

The other required definitions, assumptions, and notations are:

$p(X_k|Z_k)$ = a posteriori probability density function (PDF) associated with a candidate trajectory, given all the measurements Z_k.

w_k = random forcing function at the k-th stage; w is assumed white and independent of x_0.

v_k = measurement noise. v is white and independent of w and x_0.

Using the above notation, we can formulate the general optimum estimation problem as follows [1][2]. Given a known difference equation describing the dynamical system

$$x_{k+1} = f(x_k, w_k) \ , \tag{5.1}$$

the a priori PDF of the initial states $p(x_0)$, the statistics of v, $p(v_i)$, and w, $p(w_i)$, i=1,...,k , the measurements z_i, i=1,...,k , and the known measurements relationship

$$z_k = h(x_k, v_k) \ , \tag{5.2}$$

find the sequence of states (trajectory), x_k, that maximizes the conditional PDF

$$p(X_k|Z_k) \ . \tag{5.3}$$

Notice that, in our case, (5.1) describes a simple straight-line extension of states (or trajectory segements), where w_k is the pixels quantization noise, and (5.2) describes the multi-sensor Spectrogram measurements, where v_k is the measurements noise.

The basic idea of the proposed algorithm is to compute the likelihood function for all possible trajectories, and choose those having sufficiently high likelihood as candidate detections. To see more clearly the factors that enter this function let us consider what happens when we advance by one stage. Using Bayes' theorem it can be shown that

$$p(X_{k+1}|Z_{k+1}) = \frac{p(z_{k+1}|x_{k+1})p(x_{k+1}|x_k)}{p(z_{k+1}|Z_k)} p(X_k|Z_k) \tag{5.4}$$

(see [1]-[5]). The function $p(z_{k+1}|Z_k)$ is used for normalization purposes only and need not be carried along.

$p(z_{k+1}|x_{k+1})$ is the PDF of the measurements at the k+1-st stage, given that a particular state, x_{k+1}, contains a target. This PDF takes into account the intensity of the return in the time/frequency pixels associated with the given state and information about target signature. The probability will be higher if the signature of the return associated with state x_{k+1} looks like a target, and lower if it does not have typical target characteristics. This term will be in our case the output of the x_{k+1} matched filter.

$p(x_{k+1}|x_k)$ is the PDF of a target being in state x_{k+1}, given that it was previously in state x_k. It is here that we incorporate into the algorithm information about target dynamics. Since a target can change its course, speed and location only gradually, high probability will be given to a state x_{k+1} consistent with previous target parameters (course/speed/location), and low probability will be given to states corresponding to large changes in target parameters.

We now split the required maximization of (5.3) as follows:

$$\max_{X_{k+1}} p(X_{k+1}|Z_{k+1}) = \max_{x_{k+1}}\left[\max_{X_k}\{p(X_{k+1}|Z_{k+1})\}\right] . \tag{5.5}$$

Using (5.4) and defining the Merit Function, $I(x_{k+1})$, as the inner maximum on the righthand side of (5.5), we can arrive at the recursion (see [2] and [13]-[16])

$$I(x_{k+1}) = p(z_{k+1}|x_{k+1}) \max_{x_k}\{p(x_{k+1}|x_k)I(x_k)\} \tag{5.6}$$

which defines a first-order Markov relationship. Initially, $I(x_o) = p(x_o)$ is the (equal) a priori PDF for all the states at stage zero. From (5.6) we see that for each state, x_{k+1}, there is a certain $\hat{x}_k$ that yields its maximal $I(x_{k+1})$. This functional relationship can be expressed as

$$\hat{x}_k = \hat{x}_k(x_{k+1}) . \tag{5.7}$$

The algorithmic interpretation of equations (5.6) and (5.7) is the following.

a) Define a close set of discrete states (1-minute straight trajectory segments) or alphabet which are sufficient to fully relate the state of the "system" at any given time (or stage).

b) Define the discrete stages (time intervals) so that the system advances from one stage to the next stage.

c) Define the discrete Merit (or cost) Function for all states as in (5.6).

d) Start with any constant Merit Function at stage zero and calculate (5.6) for all allowed states at the next stage.

e) At each stage replace the old Merit Function, $I(x_k)$, by the updated one, $I(x_{k+1})$, and store the optimal $\hat{x}_k$ of (5.7).

f) The end reslt at the last stage is the (single) overall Merit Function and the sets of $\hat{x}_k$ for all stages. This is the point at which we finally perform the (last) outside maximization of (5.5) and find the single state having the largest Merit Function. This is the last state of the optimal trajectory in the case where a single target is known to exist.

g) Now go backward from this optimal last state and use the stored $\hat{x}_k$ data to retrieve the best previous state leading into the optimal x_o . This process is called "retrieval" because, starting from the best last state, it is possible to read off the complete optimal trajectory back to the first stage. In fact it is possible to read off the whole trajectory leading to <u>any</u> state at the last (or other) stage even if the state is not the one yielding <u>the</u> maximum Merit Function. Thus, in our algorithm we do not perform the above last optimization and, instead, we retrieve <u>all</u> trajectories whose final Merit Function crossed some threshold. This modification allows for

cases where it is not known in advance how many actual trajectories (if at all) are present in the ocean volume of interest.

The process described above can be further explained by reference to Figures 5.3 to 5.5. Figure 5.3 shows how the initially constant Merit Function evolves throughout the stages until it manifests a distinct peak (for a single-target case). The last optimal state of the trajectory in this example is $\hat{x}_k = st_{n-6}$, where n is the total number of states in the alphabet. The "signal" to sidelobes "noise" can be used to measure our confidence in the detection process. Figure 5.4 shows how the DPA proceeds forward while looking backwards from each state to determine its optimal previous state. Figure 5.5 shows the retrieval process for a simple 10-state case. States #2 and #9 are retrieved here.

5.2 GENERALIZING THE DPA

The previous section explained the DPA application to the single-frequency multi-target case through the use of the Spectrograms obtained from several sensors. The main power of the DPA is that the very same general formulation can take into account all the given a priori data and measurements. For example:

- the term $p(x_o)$ can relate any a priori statistical knowledge about the initial targets' parameters,

- the choice of the alphabet of states, x, can incorporate any known location and direction bounds in addition to the already accounted-for speed bounds,

- the $p(z_k|x_k)$ term (the output of the multi-sensor Spectrograms matched-filters) can incorporate other known frequencies or broadband sources as well as Correlogram and Doppler Spectrograms measurements.

In summary, the optimality of this algorithm (assuming Normal measurements noise) results from the application of matched-filtering to the

data, and from the incorporation of all the a priori intelligence or other data through the DPA. The DPA also makes this optimal detection/tracking feasible in terms of computations/memory requirements.

5.3 COMPARISON WITH OTHER TRACKING ROUTINES

The "Maple" line-tracking routine [17] is an example of a "diminished-scale" DPA application to the line tracking problem. It uses matched-filtering like we do, but it tracks frequency trajectories defined in the Spectrogram image and not physical submarine (x,y,z) trajectories defined in the ocean volume. Although this algorithm is optimal for what it is doing, it is far suboptimal from our viewpoint. First, it does not start frequency-line trajectories everywhere, and second, more important, it does not incorporate the signals from all sensors and all "Grams." Further, it cannot incorporate a priori knowledge about the expected trajectory parameters or bounds.

The Maple-algorithm excellent performance can, however, serve as a precursor to the performance which is achievable by using our suggested full-scale DPA.

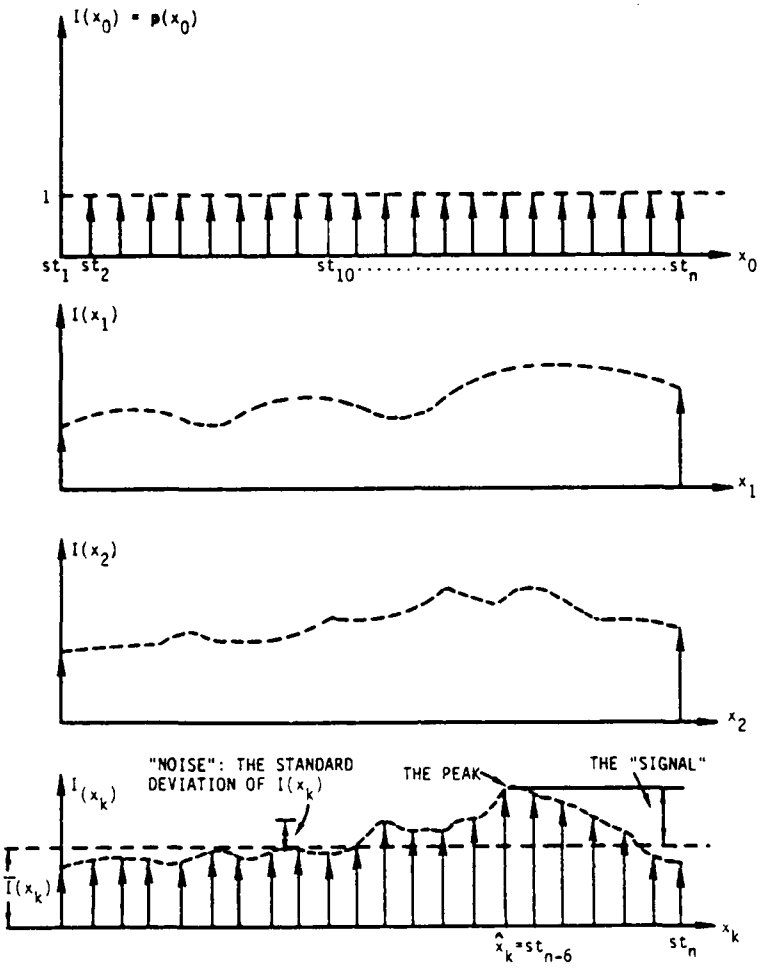


Figure 5.3. The Evolvement of the Discrete PDF During the Iterations

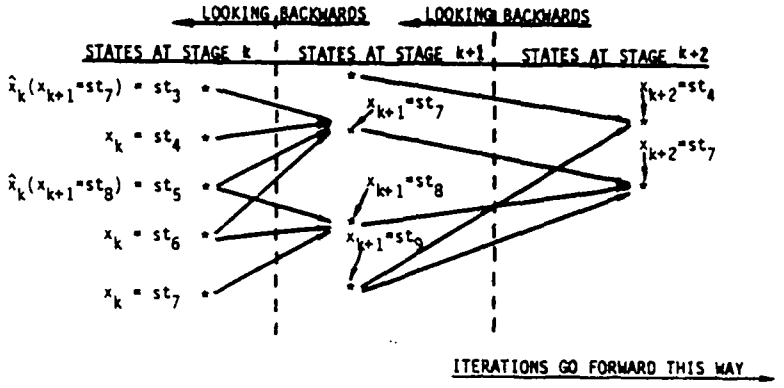


Figure 5.4 The Iteration Process

x_0	$\hat{x}_0(x_1)$	$\hat{x}_1(x_2)$	$\hat{x}_2(x_3)$	$\hat{x}_3(x_4)$	$\hat{x}_4(x_5)$	$\hat{x}_5(x_6)$	$\hat{x}_6(x_7)$	$\hat{x}_7(x_8)$	$I(x_8)$
1	5	3	2	(9)	(1)	5	5	1	0.41
(2)	3	2	1	(7)	8	7	(4)	5	0.38
3	4	5	8	4	4	3	3	(2)	(0.95)
4	(2)	2	10	3	(2)	(1)	2	8	0.31
5	1	(4)	7	10	9	8	1	4	0.11
6	8	10	4	8	7	2	7	3	0.23
7	7	8	(9)	1	9	3	8	9	0.39
8	8	7	4	4	6	5	10	(9)	(0.88)
(9)	10	(10)	(5)	9	7	(4)	(9)	7	0.62
10	(9)	3	8	7	8	10	7	2	0.32

Figure 5.5 The Retrieval Process

References for the Dynamic Programming Approach to Detection and Tracking (Section 5)

1. Bellman, R., "Dynamic Programming," Princeton, New Jersey: Princeton University Press, 1957.

2. Larson, R.E., and J. Peschon, "A Dynamic Programming Approach to Trajectory Estimation," IEEE Transactions on Automatic Control, AC-11, 3, 537-540, July 1966.

3. Scharf, L.L., and H. Elliott, "Aspects of Dynamic Programming in Signal and Image Processing," IEEE Transactions on Automatic Control, AC-26, 5, 1018-1029, October 1981.

4. Kalman, R.E. and R.S. Bucy, "New Results in Linear Filtering and Prediction Theory," Transaction ASME J. Basic Engieering, Vol. 83, Series D, No. 1, pp. 95-108, March 1961.

5. Viterbi, A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Transactions on Information Theory, Vol. IT-13, No. 2, pp. 260-267, April, 1967.

6. Larson, R. E., "A Survey of Dynamic Programming Computational Procedures," IEEE Transactions on Automatic Control, Vol. AC-12, No.6, pp. 767-774 December, 1967.

7. Larson, R.E., and J.L. Casti, Principles of Dynamic Programming, Part II, Advanced Theory and Applications, Marcel Dekker, Inc., New York, 1982.

8. Larson, R.E. L. Meier, and A.J. Korsak, "Optimal Filtering with Quantized Data," Joint Automatic Control Conference, University of Michigan, Ann Arbor, Michigan June, 1968.

9. Tse, E. and R.E. Larson, "Parallel Algorithms for Optimum Nonlinear State Estimation," Joint Automatic Control Conference, Ohio State University, Columbus, Ohio June, 1973.

10. Larson, R.E. and E. Tse, "Parallel Processing Algorithms for the Optimal Control of Nonlinear Dynamic Systems," IEEE Transactions on Computers, Vol. C-22, No. 8, pp. 777-786, August, 1973.

11. Larson, R.E. and J.L. Casti, Principles of Dynamic Programming, Part I, Basic Analytic and Computational Methods, Marcel Dekker, Inc., New York, 1978.

12. Larson, R.E., R.M. Dressler, and R.S. Ratner, "Precomputation of the Weighting Matrix in an Extended Kalman Filter," JACC, Univ. of Pa., Philadelphia, Pa., June 1967.

13. Barniv, Y., "Dynamic Programming Algorithm for Mosaic Sensor Tracking," Final Report, DARPA/NOSC Contract No. N66001-82-C-0271, Systems Control Technology, Inc., Palo Alto, CA, February 1983.

14. Barniv, Y., "Application of the Dynamic Programming Algorithm to IR Sensor

Streak Detection," Technical Report, DARPA/NOSC Contract No. N66857-83-C-1015, Systems Control Technology, Inc., Palo Alto, California, October 1983.

15. Barniv, Y., "Dynamic Programming Algorithm for Mosaic Sensor Tracking," Final Report 5492-02, Systems Control Technology, Inc., Palo Alto, California, November 1984.

16. Barniv, Y., "Dynamic Programmming Solution for Detecting Dim Moving Targets," IEEE Trans. Aerospace Electronic Systems, Vol. AES-21, No. 1, pp. 144-156, January 1985.

17. Wolcih, Joseph J., "Maximum A Posteriori Estimation of Narrow-band Signal Parameters," Journal Acoustic Society of America, 68(1), July 1980, pp 174-178.

7. PROPOSED STATEMENT OF WORK

7.1 INTERACTIVE SIGNAL PROCESSING WORKSTATION

The workstation project will consist of the following tasks:

Task 1: Library Development

The large collection of existing Fortran and CtrlC signal processing utilities needs to be organized into one package. A uniform style of documentation is needed.

Task 2: Module Interface Development

Many of the signal processing modules need a "generic interface" which is called interactively by the user. An interface routine provides (for example) argument defaulting, argument checking, dynamic allocation, and automatic type conversions when necessary. Currently, such modules are used to install Fortran subroutines in CtrlC.

Task 3: CtrlC Language Extensions

Numerous new language facilities are planned for CtrlC to further enhance its expressive power. For example, the macro capability will be made much more powerful. In addition, new high-level debugging aids are planned.

Task 4: CtrlC Environment Extensions

This task includes development of all general display and user interface capabilities. To harness similar existing capabilities as much as possible, CtrlC will be ported to a high quality interactive programming environment which supports multiple process windows. The necessary CtrlC specific enhancements will then be installed.

Task 5: CtrlC HELP Extensions

This task involves installing the user-directed documentation facility described in this paper. In essence the proposed help facility is a particularly easy to use relational database.

Task 6: CtrlC to Fortran Compiler

It is desired to be able to translate the CtrlC high-level language into highly portable Fortran.

Task 7: Block-Diagram User Interface

The generic block diagram capability is being developed in a separate project. This task involves fully integrating the block diagramming facility into the system environment.

Task 8: The Integrated Editor

Hopefully the desired editor will be largely provided with the target interactive programming environment. Even so, some customization of the editor will be desirable. Integration with the block diagram facility is desired.

7.2 DYNAMIC PROGRAMMING FOR MULTITARGET DETECTION AND TRACKING

The DP algorithm has been developed at SCT for the problem of detecting/tracking aircraft from space as they are observed in an IR imagery set. The algorithm development got to the point of successful demonstration by simulation, yielding results which are in excellent agreement with the theoretical performance estimates. In the following we suggest to develop a similar algorithm for the ocean underwater surveillance problem along the same lines that led us to the existing algorithm.

Task 1: Algorithm Development

We will develop the Dynamic Programming Algorithm for the underwater surveillance problem as outlined in the preceding text. This will include defining states, stages, transitions and measurements functions, and the Merit Function.

Task 2: Analysis

The purpose of the analysis effort is to optimize the algorithm and to predict its performance in term of probabilities of detection and of false alarm. Optimization of the algorithm will include tradeoffs between performance and computational cost.

Task 3: Simulation

The purpose of the simulation work is to verify and complement the analysis. Each of the analytical results need to be validated by simulation. Also, various cases whose performance is difficult to analyze, need to be evaluated by simulation. An example of such a case is that of targets which exit or enter the relevant ocean volume during the current K-frame processing interval.

Appendix A

In this appendix, the result of the CtrlC "BROWSE" command is listed. The BROWSE command lists all major functions in the CtrlC environment organized by topic.

A.1.0 CTRL-C INFORMATION ADN ASSISTANCE

browse - displays this CTRL-C command synopsis list
help - help facility
intro - a quick introduction for the new user
logo - graphs CTRL-C logo
news - news on updates and new additions
what - lists current User-Defined Functions and libraries
who - lists current variables
why - gives succinct answers to any questions

A.2.0 STARTING, INTERRUPTING, AND ENDING THE CTRL-C SESSION

exit - end CTRL-C session and save current workspace
quit - end CTRL-C session and do not save current workspace
resume - resume CTRL-C session with workspace from last session
$ - interrupts the session and allows a VMS command to be typed
↑Y - aborts CTRL-C
↑C - local CTRL-C abort, stops current command

A.3.0 BASICS

A.3.1 SETTING THE OPERATING ENVIRONMENT

char - character set replacement
chop - truncate arithmetic
clear - erase variables, functions, or libraries from work space
disp - displays variables containing text
hard - specifies the hardcopy device type for graphics
lines - sets maximum number of lines of a matrix that are displayed

long - sets long numerical output display format
page - clears alphanumeric CRT screen
semi - cause or suppress printing of operation results
short - sets short numerical output display format
term - specifies the terminal type for graphics

A.3.2 SPECIAL SYMBOLS

[- used in forming vectors and matrices (also used for macros)
] - see [
< - means "get input from file" or less than
> - means "send output to file" or greater than
(- indicates precedence in arithmetic expressions in the usual way
) - see (
= - used in assignment statements
. - Decimal point. Element-by-element operations. Kronecker products.
.. - continuation of a statement onto the next line
, - separates matrix subscripts and function arguments
; - Ends rows. Suppresses printing of commands.
/ - matrix right division computed by Gaussian elimination
// - comment statement, function definition
\ - matrix left division computed by Gaussian elimination
' - matrix transpose, quote to delimit character strings
+ - matrix addition
- - matrix subtraction
* - matrix multiplication
** - raises matrices to powers
: - used in subscripts and FOR iterations
$ - interrupts the session and allows a VMS command to be typed

A.3.3 SPECIAL PERMANENT VARIABLES

ans - variable created when expressions are not assigned
eps - floating point relative accuracy
flop - count of floating point operations
hard - specifies the hardcopy device type for graphics

pi - 3.1415926536
term - specifies the terminal type for graphics

A.3.4 PROGRAM FLOW STATEMENTS

else - used with if
end - delineates 'for', 'while', and 'if' statements
exit - terminates a 'for' or 'while' loop; or ends session
for - repeat statements a specific number of times
if - conditionally execute statements
while - repeat statements while an expression is true

A.4.0 GRAPHICS

awin - split screen or window on terminal in alphanumeric mode
disp - compact matrix print as +, -, and blank
erase - erases entire plotting surface
page - clears alphanumeric CRT screen
plot - X-Y point, bar and line plots
pline - GKS polyline primitive
pmark - GKS polymarker primitive
p3d - 3-dimensional plotting of a matrix
replot - repeat plot to a hardcopy file
redterm- redirect terminal graphics to a file
redhard- redirect hardcopy graphics to a file
text - GKS text placement primitive
title - plot titles for PLOT and P3D
window - specifies the window for subsequent graphics commands
xlabel - X-axis labels for PLOT
ylabel - Y-axis labels for PLOT

A.5.0 PROCEDURES AND USER-DEFINED FUNCTIONS

ascii - converts variables from decimal to ASCII
base - converts numbers to different bases
echo - echoing of the commands

emode - stops CTRL-C if an error occurs
do - runs a CTRL-C procedure located in a file
disp - displays variables as character strings
edf - edit a function using the EDT editor
deff - define a procedure to be a function
glob - defines global variables within functions
input - prompts user for input
str - Number to string conversion
lib - function library maintainer
pause - pause until <cr> entered
return- returns from a user function

A.6.0 DISK FILES

diary - saves a "diary" of the session in a disk file
key - saves all keyboard input in a file
load - loads variables from a file
print - prints variables on a file with 132 columns
save - saves variables on a file

A.7.0 MATRIX ANALYSIS

A.7.1 ELEMENT BY ELEMENT MATRIX FUNCTIONS

abs - absolute value or magnitude
conj - complex conjugate
imag - imaginary part
max - largest value
min - smallest value
prod - product of elements
real - real part
round - round to nearest integer
sum - sum of the elements of a matrix

A.7.2 SQUARE MATRIX ELEMENTARY FUNCTIONS

atan - arctangent
cos - cosine
exp - matrix exponential
log - natural logarithm
inv - inverse of a matrix
sin - sine
sqrt - square root

A.7.3 MATRIX PROPERTIES

cond - condition number in 2-norm
det - determinant
norm - singular values, 1-norm, infinity norm, and F-norm
rank - rank of a matrix
rat - remove roundoff error
rcond - estimate of the condition of a matrix

A.7.4 MATRIX DECOMPOSITIONS AND FACTORIZATIONS

chol - Cholesky factorization
eig - eigenvalues and eigenvectors
geig - generalized eigenvalues using the QZ algorithm
hess - Hessenberg form
lu - factors from Gaussian elimination
orth - orthogonalization
qr - orthogonal-triangular decomposition
schur - Schur decomposition
rref - reduced row echelon form of a rectangular matrix
svd - singular value decomposition

A.7.5 MATRIX DIFFERENTIAL EQUATION SOLUTIONS

are - algebraic Riccati equation solution
dare - discrete algebraic Riccati equation solution

lyap - Lyapunov equation solution
dlyap - discrete Lyapunov equation solution

A.7.6 OTHER OPERATIONS

conv - convolution and polynomial multiplication
deconv- deconvolution and polynomial division
pinv - pseudoinverse
poly - characteristic polynomial
roots - find polynomial roots
kron - Kronecker tensor product
size - row and column dimensions of a matrix
sort - sorting
less - less than or greater than

A.7.7 MATRIX BUILDING FUNCTIONS

diag - puts in/ takes out vectors on matrix diagonals
eye - generates identity matrices
hilb - generates an inverse Hilbert matrix
magic - generates a magic square
ones - generates a matrix of ones
rand - generates random numbers and matrices
tril - lower triangular part
triu - upper triangular part

A.8.0 CONTROL DESIGN AND ANALYSIS

A.8.1 CONTINUOUS-TIME SYSTEMS DESIGN

s = are(a,b,q,r)	algebraic Riccati equation solution
[k,p] = lqe(a,b,c,q,r)	optimal LQG estimator design
[k,s] = lqr(a,b,q,r)	optimal LQG regulator design
[k,s] = lqry(a,b,c,d,q,r,n)	LQG regulator design with output weighting
[k,s] = impl(a,b,c,f,q,r)	implicit model following gain calculation

```
k      = place(a,b,p)          regulator and estimator pole placement
                               (SISO)
r        rloc(n,d,k)           root loci Analysis Time domain
    y = impul(a,b,c,iu,t)      impulse response
    y = step(a,b,c,d,iu,t)     step response
    y = ramp(a,b,c,d,iu,t)     ramp response
    y = simu(a,b,c,d,u,t)      simulation with arbitrary inputs Frequency
                               domain
[m,p] = bode(a,b,c,d,iu,w)     Bode and Nichols frequency response
[r,i] = nyqu(a,b,c,d,iu,w)     Nyquist frequency response
g      = freq(a,b,c,d,s)       complex frequency response
sv     = sigma(a,b,c,d,w)      singular value frequency response (sigma
                               plot)
w      = logspace(d1,d2)       frequency vector generation Conversions
c2d    - continous to discrete state space
c2dt   - continuous to discrete state space with pure delay
 psit  - intermediate discrete conversion function
ss2tf  - state space to Laplace tranfer function
tf2ss  - Laplace transfer function to state-space
```

A.8.2 DISCRETE-TIME SYSTEMS DESIGN

```
s      = dare(a,b,q,r)         discrete algebraic Riccati equation
[k,p] = dlqe(a,b,c,q,r)        optimal LQG estimator design
[k,s] = dlqr(a,b,q,r)          optimal LQG regulator design
x      = dlyap(f,q)            discrete Lyapunov equation solution
r      =  rloc(n,d,k)          root loci Analysis Time domain
y      = dsimu(phi,gam,c,d,u)  simulation with arbitrary inputs
                               Frequency domain
[m,p] = dbode(a,b,c,d,iu,w)    Bode and Nichols frequency response
g      = freq(a,b,c,d,s)       complex frequency response
w      = logspace(d1,pi)       frequency vector generation Conversions
ss2tf  - state space to Z-transform
tf2ss  - Z-transform to state-space
```

A.8.3 OTHER BUILDING STATE-SPACE SYSTSEMS

series - series connection
paral - parallel connection
interc - block diagram system interconnection
System Matrix properties
ctrb - controllability staircase form
obsv - observability staircase form
minreal - minimum realization
balreal - balanced realization
gram - controllability and observability gramians
stair - staircase algorithm
tzero - transmission zeros

A.9.0 DIGITAL SIGNAL PROCESSING

conv - convolution
deconv - deconvolution
fft - 1 and 2 dimensional FFT
ifft - 1 and 2 dimensional inverse FFT
tdlf - tapped-delay-line filter
maxlike - maximum likelihood identification
rml - recursive maximum likelihood for ARMAX model

Typing 'BROWSE >filename' will send this listing to a file where you may print it on your local hardcopy device.

Appendix B

In this appendix, the currently existing version of CtrlC is used in a simple filter design exercise:

```
// Example:  Design of an FIR bandpass filter using windowing

// Step 1: Specify desired frequency response over the first
//         half of a 512 point sequence:

Hfd = [0*ones(1,25) ones(1,20) 0*ones(1,212)]; // [000011110000] (Bandpass)

// Step 2: Reflect about point 1 (mod 512) for the remainder of the
//         sequence. The frequency specification must be symmetric
//         in order to have real filter coefficients.

Hf = [Hfd Hfd(256:-1:2)];   // Desired magnitude frequency response

ht = ifft(Hf);   // Step 3: Create corresponding impulse response

// Step 4: Window the impulse response for shorter filter length

NN = 50;n = 1:nn;
 w = 0.5*(ones(n) - cos(2*pi*n/(NN-1)));   // Use a Hanning window
 b = [ht(488:512) ht(1:25)].*w;

// Step 5: Find actual frequency response

ht  = [b 0*ones(1,512-NN)];
Hf2  = fft(ht); // True complex frequency response

Overlay(Db(Hf),Db(Hf2)); // Plot overlay of desired and obtained F.R.
```

AUTHORS' BIOGRAPHIES

JULIUS SMITH, Research Engineer

Education

Julius Smith received the B.S. degree (Magna Cum Laude) in Electrical Engineering from Rice University in 1975, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1978 and 1983, respectively.

Specialization

Dr. Smith maintains professional specialization in several areas of digital signal processing. His most recent work has been in the areas of adaptive filtering, system identification, digital filter design, delay estimation, spectrum analysis, and signal-processing system development.

Background

Dr. Smith entered Rice University on a Moody Foundation scholarship, and, while at Rice, received the Brown Engineering Merit Award in two successive years. In his Junior year he was elected to Tau Beta Pi. In 1975 he finished his Bachelor's degree (Magna Cum Laude) in electrical engineering, specializing in control, communications, and circuits. Julius was fortunate to have taken every course taught by Prof.'s C. S. Burrus and J. B. Pearson. This provided him with a strong foundation for further research in digital signal processing.

During his last year at Rice, Julius worked for the IBM Houston Scientific Center codesigning and implementing a user interface to a comprehensive set of tools for power systems analysis. The software package was published under the acronym MAPS for Modular Analysis of Power Systems.

Immediately after graduation from Rice in 1975, Dr. Smith began working as a signal analyst in training at the Electromagnetic Systems Laboratories

(ESL) in Sunnyvale California.

Dr. Smith's first project at ESL was to design and simulate the digital AM/FM subsystem of an automatic signal recognition system. He was responsible for high-level hardware specifications and system performance evaluation. In connected tasks, he developed new signal detection techniques to exploit advantages inherent in his AM/FM receiver design. During this time Dr. Smith acquired extensive experience in applied digital signal processing.

In 1977, Dr. Smith was awarded a Hertz Foundation Graduate Fellowship, and in the Fall of that year he became a full-time graduate student at Stanford in the Electrical Engineering Department. The first year was devoted to completing course work for the Master's degree, and he received his M.S. in 1978 with a near perfect GPA.

In the summer of 1978 Dr. Smith returned to ESL to implement the dynamic-programming component of an automatic language discriminator. The discriminator produced a forced decision among five different languages based on the likelihood of the estimated phoneme transition probabilities.

Also in the summer of 1978, Dr. Smith performed consulting work for Total Technology Inc. His task was to model optical transfer functions of phosphor screens used in image-intensifier tubes. This was Dr. Smith's first involvement with the field of system identification which became a major emphasis of his thesis work.

In the summer of 1979, Dr. Smith worked in the Acoustics Research Department of Bell Laboratories in Murray Hill New Jersey. With Jont B. Allen, he designed, implemented, evaluated, and published a new technique for adaptive delta modulation (ADM). The new ADM coder operated with a variable sampling rate which was slaved to the measured input signal bandwidth.

In the summer of 1980, Dr. Smith returned to ESL to carry out research applying linear prediction techniques to FSK demodulation. Two novel algorithms were developed, evaluated, and documented in an internal final report.

Also in the summer of 1980, Dr. Smith performed consulting services for Narendra Gupta of Integrated Systems Inc. His task was to implement a maximum-likelihood system identification program which was based on a state-space model explicitly parametrized in terms of second-order mode frequencies, bandwidths, gains, and initial conditions.

In the summer of 1980, Dr. Smith took a brief visiting position at the Computer Audio Research Laboratory, University of California, San Diego. There he installed the IEEE Programs for Digital Signal Processing on a Vax 11/780 mainframe computer and produced user-interface software in the C programming language which made extensive use of the Unix pipes facility. The result was an elegant and powerful set of signal processing tools which continue to be widely distributed by UCSD in the public domain.

By 1980 Dr. Smith had completed his Ph.D. course work at Stanford, and he devoted the next two years to his thesis research. The first area of intense study was system identification. During this time Prof. Lennart Ljung was a visiting professor at Stanford, and Dr. Smith benefited greatly from pursuing problems under Prof. Ljung's direction. In addition to developing techniques for system identification, Dr. Smith developed several contributions to digital filter design (a closely related area). His work in both areas concentrated on effective manipulation of the error criterion minimized by modeling techniques. Some of the most useful improvements were obtained through the use of conformal mapping techniques and other preprocessing steps. He proved that there is no upper bound on the number of false local minima when minimizing output error with respect to even a single filter pole. His thesis contains what is perhaps the first use of Hankel norm theory to obtain a globally convergent algorithm for optimally approximating a complex desired frequency response with a recursive digital filter. The Ph.D. degree was awarded in June 1983. Later that year he was elected to the Sigma Xi society.

In the Fall of 1982, Dr. Smith joined the Adaptive Systems Department of the Advanced Technology Division of Systems Control Technology (SCT) in Palo Alto California. He worked half time until December 1982 whereupon he became a full time SCT employee. He continued full time until October of 1984 at

which time he reduced his commitment to 60 percent in order to make time to teach a two-year course sequence in digital signal processing at Stanford. Dr. Smith has worked exclusively on projects directed by Dr. Benjamin Friedlander at SCT for the past three years.

Tasks at SCT

The overall theme of recent research in the Apdaptive Systems Department at SCT has been to develop modern algorithms which address problems fundamental to target localization based on multiple hydrophone recordings. The principal new technologies brought to bear on the localization objective have been drawn from the most recent advances in system identification and parametric spectral estimation. Complementing this line of research, Dr. Smith has developed more traditional sonar signal processing facilities for nonparametric spectral estimation based on the FFT.

The first project assigned to Dr. Smith in 1983 was to develop an adaptive time-delay estimation algorithm using fundamental formulations from recursive system identification. This effort resulted in a highly efficient and accurate method with a very rapid tracking capability in the time-varying case. The heart of the method was a novel intersample interpolator developed by Dr. Smith for his thesis applications.

The adaptive delay estimation technique was next extended in two straightforward ways to the problem of adaptively tracking multipath delay. The techniques were compared in a precise way to the maximum likelihood estimator for the multipath estimation problem. Theoretical characterizations of the asymptotic properties of the estimates were carried out, including global convergence studies, derivation of asymptotic parameter variance (with comparisons to the Cramer-Rao lower bound), analysis of multipath detection/false-alarm probabilities, and the effects of multipath on target localization.

A third task was to develop an adaptive notch filter algorithm, again applying the latest results in recursive identification as well as some new theorems on rational transfer functions developed by Dr. Smith in the course

of his thesis work.

A task carried out in parallel with the above identification projects was the ongoing evolution of the Modified Yule-Walker (MYW) method for ARMA spectral estimation. More recent variations of the MYW approach have been termed Optimal Instrumental Variable methods. The MYW methods can be viewed as variations of Prony's method for pole-zero digital filter design using estimates of the correlation function in place of the filter impulse response.

The numerous MYW enhancements were aimed at achieving reduced variance and bias in the parameter estimates for a given number of data observations. Dr. Smith performed all simulations and software development for these studies and contributed major implementation decisions. Very significant improvements over the original MYW technique were realized. To aid in the dissemination of these superior techniques, Dr. Smith wrote highly portable Fortran versions of the MYW methods.

A later task was to implement a technique for multiple target localization from narrowband Doppler measurements. The method consists of finding intersections between hypersurfaces in a five-dimensional space. Each target appears (in the noiseless case) as a point of intersection of all the surfaces. Dr. Smith developed novel means of analyzing the behavior of this algorithm in the presence of noisy measurements. Dr. Smith also developed a clustering algorithm for the technique, including a method for determining the number of targets, associating spectral lines to targets, and assigning points of intersection to targets. As in all projects on which he worked, Dr. Smith was responsible for all software development and computer simulations.

In another parallel series of projects, Dr. Smith developed recursive identification counterparts to the Constant Modulus Algorithm (CMA) which is a novel method of adaptive channel equalization pioneered by J. R. Treichler. Dr. Smith's enhancements included extension to a nonrecursive channel model (ideal for removing multipath distortion), a real-signal version with convergence properties as good as the complex-signal version, a quadratic Newton-method to replace fixed-step-size gradient descent, an interference rejection technique, and more. In addition, Dr. Smith first proved global

convergence of the CMA from an arbitrary starting point for the model-complete case. This was perhaps the first global convergence proof in the system identification literature which involved a prediction error quadratic rather than linear in the model parameters. The notion of "persistently exciting" was generalized to the nonsingularity of a rank four "covariance tensor."

To obtain useful benchmarks of comparison between the ARMA spectral estimation methods (MYW) and the perceived standard practice, Dr. Smith developed an implementation of the periodogram method for spectrum estimation (PMPSE) using the FPS array processor. The PMPSE essentially averages the squared magnitude of successive overlapped, Hamming-windowed, short-time FFT's. In an internal report by Dr. Smith containing the MYW and PMPSE programs, a review of the theoretical asymptotic performance of both approaches was included. Dr. Smith also developed a wide variety of spectral displays, including ordinary line plots, "waterfall diagrams," encoded gray-level laser-printer plots, and he investigated the benefits of using edge-enhancement algorithms already available at SCT.

To allow benchmark comparisons between the new techniques developed by Dr.'s Friedlander and Smith and previously existing localization systems, Dr. Smith is currently finishing up a replication of what is considered standard low-level sonar signal processing practice. Estimates of intersensor time delay, multipath time delay, intersensor differential Doppler, and absolute Doppler are all derived in one way or another from the FFT-based spectrum estimate (using the PMPSE). Both narrowband and broadband Doppler estimation techniques have been developed. Excellent differential and absolute Doppler estimates have been produced from live sonar recordings.

Partial List of Publications

[1] M. Gutknecht, J. O. Smith, and L. N. Trefethen, "The Caratheodory-Fejer (CF) Method for Recursive Digital Filter Design," IEEE Trans. on Acoust., Speech, and Signal Proc., vol. ASSP-31, pp. 1417-1426, Dec. 1983.

[2] J. O. Smith and J. B. Allen, "Variable Bandwidth Adaptive Delta Modulation," Bell System Technical Journal, vol. 60, no. 5, pp. 719--737, May-June 1981.

[3] J. O. Smith and J. B. Angell, "A Constant Peak-Gain Digital Resonator Tuned by a Single Coefficient," Computer Music J., vol 6, no. 4, pp. 36-40, 1982.

[4] J. O. Smith, "Techniques for Digital Filter Design and System Identification with Application to the Violin," Ph.D. Dissertation, Elec. Eng. Dept., Stanford University, June 1983.

[5] J. O. Smith, "Spectral Pre-Processing for Audio Digital Filter Design," Proc. 1983 Int. Conf. Computer Music, Rochester NY, Computer Music Assoc., 1983.

[6] D. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong Plucked String Algorithm," Computer Music J., vol. 7, no. 2, pp 56-69, 1983.

[7] J. O. Smith, "Synthesis of Bowed Strings," In Proc. 1982 Int. Computer Music Conf., T. Blum and J. Strawn, eds., Computer Music Assoc., San Francisco, 1983.

[8] Friedlander and J. O. Smith, "Analysis and Performance Evaluation of an Adaptive Notch Filter," IEEE Trans. on Info. Theory, vol. IT-30, pp 283-295, March 1984.

[9] J. O. Smith and B. Friedlander, "Adaptive Interpolated Time-Delay Estimation," IEEE Trans. on Aerospace and Electronic Systems, vol. AES-21, no. 2, pp. 180--199, March 1985. Also presented at Asilomar, 1983.

[10] J. O. Smith and P. Gossett, "A Flexible Sampling-Rate Conversion Method," Proc. IEEE Conf. Acoust. Sp. and Sig. Proc., vol. 2, pp. 19.4.1-19.4.2, San Diego, March 1984.

[11] J. O. Smith and B. Friedlander, "Estimation of Multipath Delay," ICASSPY-84: Proc. IEEE Int. Conf. Acoust. Speech and Sig. Poc., Paper 15.9, San Diego CA, March 19-21, 1984.

[12] J. O. Smith and B. Friedlander, "Localization of Multiple Targets from Doppler Measurements," and High Resolution Spectral Estimation Programs," Tech. Rep. 5466-05, Systems Control Tech., May. 1984.

[13] J. O. Smith and B. Friedlander, "Analysis and Extensions of the Constant Modulus Algorithm," Tech. Rep. 6502-01, Systems Control Tech., Oct. 1984.

[14] J. O. Smith and B. Friedlander, "Simulation Results for Extensions of the Constant Modulus Algorithm," Tech. Rep. 6502-02, Systems Control Tech., Oct. 1984.

[15] J. O. Smith and B. Friedlander, "Extensions of the Constant Modulus Algorithm," Asilomar, 1984.

[16] J. O. Smith and B. Friedlander, "Global Convergence of the Constant Modulus Algorithm," Asilomar, 1984.

[17] J. O. Smith, "Introduction to Digital Filter Theory," In J. Strawn, ed., Digital Audio Signal Processing: An Anthology. William Kaufmann, Inc., Los Altos, California, 1985.

[18] J. O. Smith, "A New Approach to Digital Reverberation using Closed Waveguide Networks," Proc. 1985 Int. Conf. Computer Music, Vancouver Canada, Computer Music Assoc., 1985. Music Dept. Tech. Rep. STAN-M-31, Stanford University, July 1985.

[19] J. O. Smith and B. Friedlander, "Adaptive Multipath Delay Estimation," IEEE Trans. on Acoust., Speech, and Signal Proc., vol. ASSP-33, pp. 812-822, Aug. 1985.

YAIR BARNIV, Senior Engineer

Dr. Barniv received his B.S. degree in Electrical Engineering from Israel-Institute-of-Technology in 1963. He received his M.S. and Ph.D. degrees in communications, stochastic processes, and signal/image processing from Carnegie-Mellon University (CMU) in 1978 and 1981, respectively. His Ph.D. thesis work was on "Multi-Sensor Image Registration".

Dr. Barniv served as a commissioned officer in the Israeli Air-Force from June 1963 until May 1968, working in maintenance and development of airborne and ground radars, and fire control systems. He was also responsible for air traffic and automatic landing instrumentation (GCA).

From May 1968 to September 1977, Dr. Barniv was with Israel Aircraft Industries (IAI). Until January 1970, he served as an electrical and control engineer designing a ground-based navigation system. From January 1970 to April 1971, he worked as a control engineer and designed the control loops for a missile, including the autopilot, the guidance, and the homing laws and logic. He also conducted air-tunnel measurements and the necessary aerodynamic data extraction.

Between April 1971 and January 1974, Dr. Barniv was with a Systems Engineering Group, whose tasks included performance estimation, feasibility, and cost-effectiveness studies in fields such as electro-acoustics, electro-optics, radar, and navigation systems.

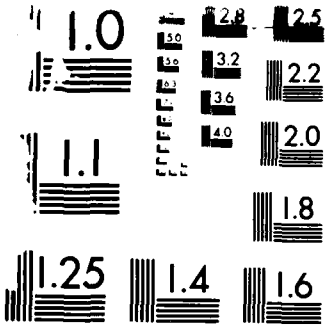
From 1974 to May 1975 he served as a senior systems engineer, heading a group tasked with preparing overall specifications for a missile design project, and coordinating the design and implementation activities and final integration. He developed the simulation programs and used them to optimize the design parameters and define the limits of performance. He was also responsible for the planning and management of the flight test program and evaluation.

From May 1975 to September 1977, he headed the Systems Engineering Group working on the control issues of the Mass Transit System for the City of Tel-Aviv. His task was to study existing systems from the point of view of automatic train control, operation, and supervision; this study included the issues of communications and control and man-machine interfaces.

From September 1977 Dr. Barniv worked on his Ph.D. thesis at Carnegie-Mellon University, and in May 1979 joined SCI as a research engineer. He is currently working in the areas of synthetic-aperture-radar, flight path optimization, closed-loop ECM techniques against various types of radars, missile's miss-distance optimization, and dynamic programming application to mosaic sensor tracking.

AD-A175 078 MULTISOURCE LOCALIZATION FROM DELAY AND DOPPLER MEASUREMENTS(U) SYSTEMS CONTROL TECHNOLOGY INC PALO ALTO CA B FRIEDLANDER ET AL. JUL 86 SCT-5503 3/3

UNCLASSIFIED N00014-83-C-0727 F/G 17/7 NL



RESOLUTION TEST CHART

Dr. Barniv's publications include:

1. "A Ground-based Navigation System," internal IAI, January 1970, (CONFIDENTIAL).

2. "A Compensation Method for the Random Effect in a Proportional Navigation System," internal IAI February 1971, (CONFIDENTIAL).

3. "A Real-time G-Bias Compensation Method," internal IAI, August 1970, (CONFIDENTIAL).

4. "Location of Guns Through the Use of Acoustical Detection," internal IAI, October 1971, (CONFIDENTIAL).

5. "Time of Arrival (TOA) Navigation," internal IAI, May 1972, (CONFIDENTIAL).

6. "The Effects of the Electrostatic Charging of an Aircraft on the On-board Instrumentation," internal IAI, March 1973, (CONFIDENTIAL).

7. "Automatic Air Raid of Naval Targets," internal IAI, October 1973, (CONFIDENTIAL).

8. "A Simple Algorithm for Fire-Zone Determination for a Ground-to-Ground Missile," internal IAI, November 1974 (CONFIDENTIAL).

9. "A Centralized Supervision System for a Mass-Transit System," internal IAI, December 1976.

10. "Intensity Disparity Compensation in Optical Pattern Recognition," M.S. Thesis, Carnegie-Mellon University, May 1978.

11. "Correlation of Images with Random Contrast Reversals," (with H. Mostafavi and David Casasent), SPIE Vol. 238, Image Processing for Missile Guidance, p. 156, 1980.

12. "Multi-Sensor Image Registration," Ph.D. Thesis, Carnegie-Mellon University, February 1981.

13. "Dynamic Programming Solution for Detecting Dim Moving Targets," IEEE Trans on AES, Vol 21, No. 3, March 1985.

Dr. Barniv's SCI or SCT publications include:

1. "Effects of Wind-Gust on the Stability of a Missile's Flight Path and the Implications on the Spectrum Spread of a Synthetic-Aperture-Radar," SCI technnical memo TM 9339-314-01, July 1979.

2. "SAR use for Anti-Tank Missiles," SCI technical memo TM 9339-314-02, July 1979.

3. "Fine Angular Control for Cruise Missiles," SCI technical memo TM 5271-006, December 1979.

END

1-87

DTIC